ESCOLA UNIVERSITARIA POLITECNICA
DE VILANOVA I LA GELTRU

BRANCA: TELECOMUNICACIO

UPC Departament de Matemàtica Aplicada i Telemàtica

SECCIÓ DE TELEMATICA

EUP Vilanova i la Geltrú

Equips Perifèrics i Terminals

PROBLEMES I SOLUCIONS

Desembre 1994

Lluís Casals i Ibáñez

UPC

UNIVERSITAT POLITECNICA DE CATALUNYA

UPC Departament de Matemàtica Aplicada i Telemàtica

SECCIO DE TELEMATICA

EUP Vilanova i la Geltrú

Equips Perifèrics i Terminals

PROBLEMES I SOLUCIONS

Desembre 1994

Lluís Casals i Ibáñez

Index

Presentació	1
Problemes proposats	2
Solucions	63

Presentació

Aquesta col.lecció de problemes és una tria de problemes de publicacions anteriors i d'examen passats, amb l'objectiu de proporcionar un material pràctic per als estudiants de l'assignatura d'Equips Perifèrics i Terminals. S'han inclòs aquells problemes que són més representatius dels conceptes bàsics i importants de l'assignatura, esperant que l'estudiant pugui exercitar-los i consolidar-los.

També es donen les solucions dels problemes proposats amb la intenció que serveixin de guia de la seva resolució o, simplement, de comprovació dels resultats obtinguts. Per això, és aconsellable que l'estudiant inverteixi un cert temps en treballar sobre el problema abans de consultar aquesta part de la publicació, fet que li reportarà un major profit en el seu estudi.

Al mateix temps, cal tenir present que alguns problemes poden tenir diverses solucions particulars i la que aquí es dóna s'ha d'entendre con un exemple de les possibles, a partir de la qual es pot concluore la generalització.

Lluís Casals Ibáñez

Vilanova i la Geltrú, desembre de 1994

Problemes proposats

PROBLEMA 1

Responeu a les preguntes que es proposen suposant que s'han executat les següents instruccions:

```
CLD
MOV AX,[100H]
MOV CX,AX
MOV SI,100H
MOV DI,150H
REP MOVSB
```

Memòria

DS:0FFh	11h
100h	22h
101h	33h
102h	44h
103h	55h
DS:104h	66h

...

a) Quantes dades (bytes) es transfereixen de memòria a memòria?

b) Valors finals de SI, DI i CX.

c) Si es volgués saber quan apareixen les adreces pel bus d'adreces, quin senyal de control es necessitaria i amb quin nivell?

PROBLEMA 2

Doneu els valors dels registres considerant que s'ha executat la instrucció en la qual apareixen com a comentari.

```
MOV AX,0004H
MOV DX,8877H
MOV CL,55H
MUL CL          ;Valor de   CL=.........
                ;           AX=.........
                ;           DX=.........
```

```
MOV AX,0000H
NEG AL
SHR AH,1          ;Valor de   AL=........
                              AH=........
AND AL,AH         ;Valor de   AL=........
                              AH=........
```

PROBLEMA 3

Considerem en aquest problema el disseny simplificat d'un rellotge digital basat en un sistema µp amb 8086.

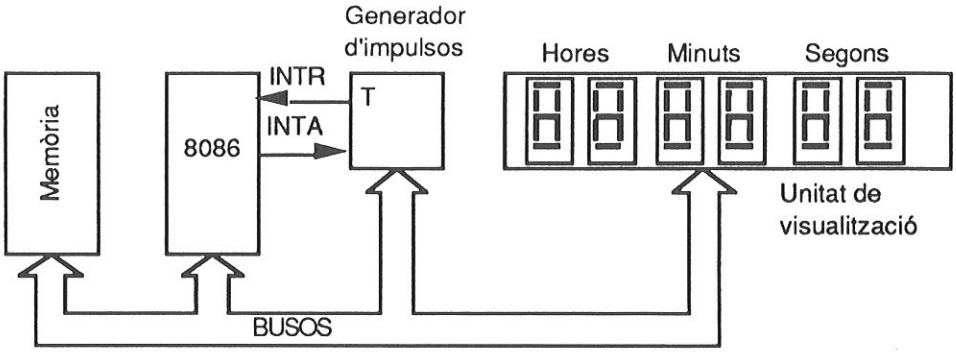


Figura 1

El sistema consta dels blocs que apareixen en la figura:

-Bloc de memòria amb el següent mapa:

4Kbytes de RAM a partir de la posició 0.

4Kbytes de ROM a partir de la posició FF000h

-µp 8086 en mode mínim.

-Generador d'impulsos: Dóna el senyal base de temporització del rellotge digital (senyal T). Aquest senyal s'aplica al terminal INTR i té una freqüència de 1000Hz.

El generador té un registre destinat a guardar el codi de tipus d'interrupció i està situat a l'adreça 21h d'E/S.

-Unitat de presentació: Consta de 6 Displays de 7 segments per a la presentació d'informació numèrica en decimal.

El programa que controla el funcionament del sistema i del rellotge digital està situat en ROM i la seva zona de dades en RAM a partir de l'adreça 400h.

a) Doneu el grup d'instruccions que hauran d'està en el programa d'inicialització del sistema que faci que les interrupcions INTR que doni el generador d'impulsos siguin del tipus 40h.

b) Si la rutina d'atenció a la interrupció que demana el generador d'impulsos ha d'estar situada a la posició FF100h, quin grup d'instruccions s'hauria d'incloure en el programa d'inicialització del sistema perquè la taula de vectors d'interrupció tingui els valors correctes?

c) Si en el moment de connectar el sistema (posar-lo en marxa) es vol que s'executi el programa d'inicialització del sistema que està situat a la posició FFA00, etiquetada com INICI, quina instrucció s'haurà de posar en la zona alta de la ROM i en quina posició exacta haurà d'està?

d) Quins senyals del 8086 s'hauran de fer servir per generar els senyals de control de lectura i escriptura de memòria i quins nivells hauran de tenir per fer aquests senyals de control?

e) El programa d'inicialització del sistema haurà de tenir un grup d'instruccions que inicialitzin la pila. Quin grup d'instruccions podria ser si la part alta de la pila, inicialment,ha d'estar situada a la posició 1000h?

PROBLEMA 4

Donat el següent programa indiqueu el valor que tindran els registres que s'especifiquen després d'executar-se la instrucció en la que apareixen com a comentari.

```
DADES SEGMENT PARA PUBLIC 'DADES'
          VAR1 DW 0FFH, 00H
          VAR2 DW 100H
               DW 023H
           ORG 100H
          VAR3 DW 0050H
               DB 25H
          TRES DB 50 DUP(3)
DADES ENDS
PILA SEGMENT PARA STACK 'PILA'
          DW 200h  DUP('PI')
PILA ENDS
SEGCOD SEGMENT PARA PUBLIC 'CODI'
          ASSUME  CS:SEGCOD,DS.DADES,SS:PILA
PRINCIP:  MOV AX,DADES          ;obtenir adreça de segment
          MOV DS,AX
          MOV AX,PILA
          MOV SS,AX
          MOV SI,VAR2
          MOV AX,WORD PTR[SI]          ; AX=...........
          MOV DI,OFFSET VAR1
          MOV AX,VAR2[DI]              ; AX=...........
```

```
                MOV BX,100H
                MOV AL,[BX][DI]          ; AL=.........
                MOV SI,0002h
                MOV AL,[BX+SI+5]         ; AL=.........
                MOV BP,06
                MOV AX,[BP]              ; AX=.........
                MOV BX,101H
                MOV AX,[BX+1]            ; AX=.........
SEGCOD ENDS
END PRINCIP
```

PROBLEMA 5

Considerem en aquest problema el disseny simplificat d'un rellotge digital basat en un sistema µp amb 8086.

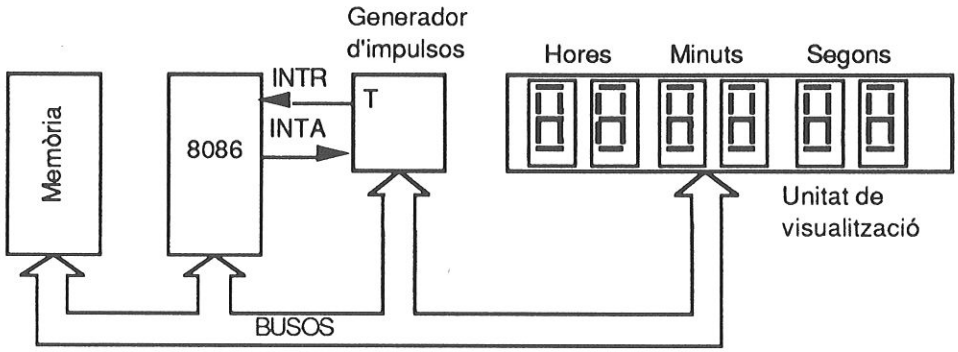


Figura 2

El sistema consta dels blocs que apareixen en la figura anterior:

-Bloc de memòria amb el següent mapa:

4Kbytes de RAM a partir de la posició 0.

4Kbytes de ROM a partir de la posició FF000h

-µp 8086 en mode mínim.

-Generador d'impulsos: Dóna el senyal base de temporització del rellotge digital (senyal T). Aquest senyal s'aplica al terminal INTR i té una freqüència de 1000Hz.

El generador té un registre destinat a guardar el codi de tipus d'interrupció i està situat a l'adreça 21h d'E/S.

-Unitat de presentació: Consta de 6 Displays de 7 segments per a la presentació d'informació numèrica en decimal. Com es veu en la figura cada display presenta una part de la informació horaria i es poden accedir com s'indica a continuació:

Display	Adreça d'E/S
Desenes hora	05
Unitats hora	04
Desenes minut	03
Unitats minut	02
Desenes segon	01
Unitats segon	00

La unitat de presentació té un descodificador de set segments per a cada display de manera que per presentar un valor en un display només cal enviar un byte amb el valor en BCD desempaquetat a l'adreça d'E/S que correspongui.

També cal tenir en compte que la unitat de presentació pot memoritzar els valors que rep. Això vol dir que només caldrà enviar informació a la unitat de presentació quan el valor del rellotge s'hagi modificat.

El programa que controla el funcionament del sistema i del rellotge digital està situat en ROM i la seva zona de dades està en RAM a partir de l'adreça 400h, tal com es pot veure en el següent esborrany del programa:

```
DADES SEGMENT AT 00000H                ;Segment situat a partir de
                                       ;l'adreça 0h
          ORG 400H
          COMPTADOR DW ?      ; paraula que actua de comptador cada
          ;cop que el 8086 rep una  interrupció 40h
          ;a través de INTR. Es decrementa aquesta
          ;paraula. Quan arriba a 0 s'ha d'incrementar
          ;el valor del rellotge digital.
;--------zona on es guarda el valor del rellotge en seg., minu. i hores----
UNISEG    DB ?        ; conté les unitats de segon
DESESEG   DB ?        ; conté les desenes de segon
UNIMIN    DB ?        ; unitats de minut
DESEMIN   DB ?        ; desenes de minut
UNIHORA   DB ?        ; unitats d'hora
```

```
DESEHORA   DB ?          ; desenes d'hora
DADES ENDS
CSEG SEGMENT PARA PUBLIC 'CODI'
            ASSUME  DS:DADES,CS:CSEG,SS:DADES,ES:DADES
INICIAL PROC FAR        ;procediment d'inicialització
            MOV AX,DADES                ; inicialització de registres de segment
            MOV DS,AX
            MOV SS,AX
            MOV ES,AX
            MOV COMPTADOR,1000          ; inicialització de la zona de dades
            MOV AL,0
            MOV DI,OFFSET UNISEG
            MOV CX,0006H
            REP STOSB
            ....
INICIAL ENDP
ATENINT PROC FAR      ; procediment d'atenció a la interrupció 40h
            .....
            IRET
ATENINT ENDP
ACTUALITZAR PROC NEAR           ; actualitza el valor del rellotge
            ......
            RET
ACTUALITZAR ENDP
TRANSFEREIX PROC NEAR           ; transfereix el valor del rellotge a la
            .....               ; unitat de visualització
            RET
TRANSFEREIX ENDP
CSEG ENDS
            END INICIAL
```

A la vista de la descripció del sistema, de la definició de dades i procediments donats:

a) Dissenyeu el procediment TRANSFEREIX. Aquest procediment ha de transferir els valors del rellotge digital al display corresponent de la unitat de visualització i serà cridat per ACTUALITZAR.

b) Dissenyeu el procediment Actualitzar. ACTUALITZAR serà cridat per ATENINT cada cop que la variable COMPTADOR=0 i haurà d'incrementar un segon el valor del rellotge digital (el valor del rellotge estarà entre 00hores 00m 00s i 23hores 59m 59s).

c) Dissenyeu el procediment ATENINT que farà de rutina d'atenció a la interrupció INTR (amb codi de vector d'interrupció 40h).

d) Suposem que les transferències del valor del rellotge digital a la unitat de presentació no les fa el 8086 sinó que és la mateixa unitat de visualització qui llegeix el valor de la memòria del sistema cada 50ms, a través d'un procés de DMA. Tot el procés de petició,concessió, transferència (de tot el valor del rellotge) i lliberació dels busos dura 300μs. Quin mode de transferència per DMA s'hauria de fer servir per tal que el procés de DMA interferís el mínim possible al funcionament del sistema?

PROBLEMA 6

En el funcionament de molts sistemes amb µp avançats es fan servir vàries piles diferents amb la intenció, per exemple, de separà dades que no pertanyen a un mateix programa. Per augmentar la seguretat d'un sistema d'aquest tipus es desitja incloure un subsistema de protecció que detecti el desbordament de la pila que el sistema està fent servir en cada moment. Per a la realització d'aquest subsitema ens basarem en el diagrama de blocs de la figura 3

La filosofia de funcionament d'aquest subsistema és la següent:

Disposa de dos registres de 16 bits (REG1 i REG2 en la figura 3) on el programa de sistema guarda l'adreça lògica (**seg:desp,** en REG1 i REG2 repectivament) del límit de la pila que es vol fer servir. Aquests registres s'han de poder accedir amb operacions d'escriptura tenint en compte que estan mapejats en E/S i que la seva entrada $\overline{CSxx}$ s'activa quan en el bus d'adreces hi ha present el valor **10H I 14H,** per REG1 i REG2, respectivament.

També conté un detector de desbordament que compara l'adreça present en el bus amb l'adreça lògica que contenen els registres REG1 i REG2 i en el cas que l'adreça del bus sigui menor es genera una interrupció mascarable activant el terminal INTR. Aquest terminal està connectat amb el terminal INTR del 8086. Degut a aquest funcionament és necessari guardar prèviament el codi de tipus d'interrupció en un registre intern del detector, que és accessible amb una operació d'escriptura sobre l'adreça **12H** d'E/S. S'ha de tenir en compte que durant un procés d'interrupció generat pel detector, aquest, necessita que s'activi la seva entrada **RI** com a senyal de reconeixement d'interrupció.

Tenint en compte el que s'ha explicat fins ara i que el 8086 del sistema es farà treballar en **mode mínim**:

a) Digeu quins senyals del 8086 s'han de connectar als terminals **A** i **B** de la figura 1.1 i dissenyar el mòdul **CTRL** si l'entrada AP és activa a nivell alt i serveix per indicar un accés a la pila.

b) Digeu quins terminals del 8086 s'han de connectar als terminals **C** i **D**, tenint en compte que els terminals STOR i STOD estan relacionats amb operacions d'escriptura. Dissenyeu també el bloc ACCES.

c) Quin senyal s'ha de connectar al terminal E?

d) Si en un punt del programa que s'estigui executant es vol canviar la pila per utilitzar una altra que està situada a l'adreça física 70010H (adreça de la primera introducció) i que com a màxim pot tenir una

longitud de 100H bytes, quina seqüència d'instruccions de 8086 s'hauria d'incloure en aquest punt del programa per fer possible el canvi i que el subsistema de protecció de la pila tingués la informació que necessita per detectar un possible desbordament?

e) Si la rutina d'atenció a la interrupció que pot generar el subsistema de protecció de la pila està situada a l'adreça lògica **8000:0100h** i si aquest subsistema ha de fer servir la interrupció tipus **82h**, quina seqüència d'instruccions de 8086 s'haurà d'incloure en la inicialització de tot el sistema (incluent el subsistema) de manera que es puguin complir aquestes especificacions?

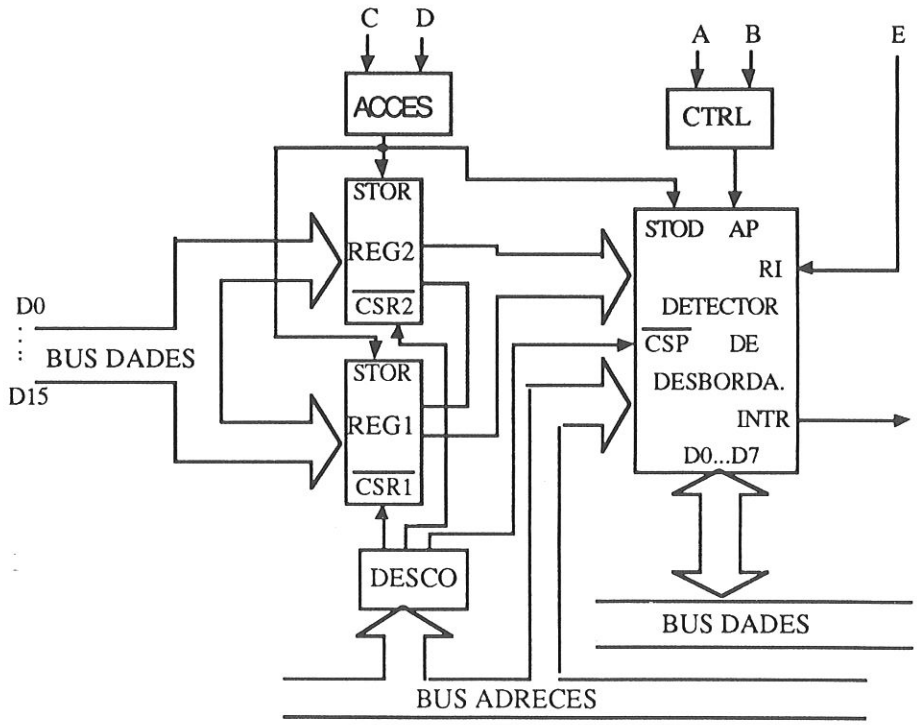


Figura 3

PROBLEMA 7

El programa següent conté dues subrutines destinades a manipular una estructura de dades, PPILA, que consisteix en una pila LIFO (Last In First Out) de byte, o sigui, que en cada accés a la pila hi ha un moviment d'un byte.

La variable PPILA, que conté l'esmentada estrutura de dades, consta de dos camps:

- La primera paraula (2 bytes) és el punter de la pila: conté la adreça de l'últim element que s'ha introduit a la pila, relativa al primer byte del segon camp (la variable ELEMENTS).

- El segon camp correspon a la variable ELEMENTS que conté els bytes que s'envien a la pila.

Una de les subrutines que hem esmentat al començament fa les operacions d'extracció (EXTREU) i l'altre les operacions d'introducció (INTROD). les operacions d'extracció i d'introducció es fan a través d'AL.

```
LONG EQU 256                    ;256=100h
CSEG SEGMENT PARA PUBLIC 'CODI'
      ASSUME CS:CSEG,DS:CSEG,ES:CSEG,SS:CSEG
      ORG 100H
ENTRADA: JMP PRINCIP            ;considereu que ocupa 3bytes
      PPILA          DW 255  ;255=FFh
      ELEMENTS       DB LONG DUP('P')

PRINCIP PROC NEAR
      ....
      MOV AL,04H
      CALL INTRO
      ....
      MOV BL,80H
      MOV AL,01H
      ADD  AL,BL
      CALL INTRO
      ....
      CALL EXTREU
      MOV BH,AL
      CALL EXTREU
      MOV AH,AL              ; (1)
      ....
PRINCIP ENDP
INTRO PROC NEAR
      MOV DI,PPILA
      DEC DI
      MOV ELEMENTS[DI],AL
      MOV PPILA,DI
      RET
INTRO ENDP
EXTREU PROC NEAR
      MOV DI, PPILA
      MOV AL, ELEMENTS[DI]
      INC DI
      MOV PPILA, DI
      RET
EXTREU ENDP
CSEG ENDS
      END ENTRADA
```

Sabent tot això respongueu a les qüestions que es plantegen:

a) Contingut de la memòria a partir de l'adreça CS:103h fins a CS:204h

b) En quin punt del programa s'executarà la directiva END ENTRADA? Què indica l'etiqueta ENTRADA?

c) Contingut dels registres BL, AL, DI en el punt (1).

d) Com afectarà a l'execució del programa un canvi d'atribut NEAR a FAR en els procediments definits en el programa?

PROBLEMA 8

Responeu a les preguntes que es plantegen, tenint en compte que el programa següent s'ha executat fins el punt referenciat en la pregunta (inclosiu) i que en la zona de memòria 40030h a 4003DH hi ha les dades inicials que es donen.

```
                MOV AX,4000H        ; inicialitza reg. segment
                MOV DS,AX
                MOV AX,5000H
                MOV ES,AX

                CLD
                MOV BX,OO30H        ;
                MOV CX,[BX]
                PUSH CX             ; guardar comptador inicial
                MOV SI,[BX+2]
                MOV DI,[BX][4]
                PUSH SI             ; (a)  guardar punter inicial
MIRAR_P:        TEST BYTE PTR[SI],01        ;comprovar si és parell
                JNZ NO_PARELL
                MOVSB               ; (b)
                LOOP MIRAR_P
                JMP IMPARELL
NO_PARELL:      INC SI              ; avançar punter
                LOOP MIRAR_P
IMPARELL:       POP SI              ; restaurar punter i comptador
                POP CX              ; (c)
MIRAR_I:        TEST BYTE PTR[SI],1 ; comprovar si és imparell
                JZ NO_IMPARELL
                MOVSB
                LOOP MIRAR_I
                JMP FI
NO_IMPARELL: INC SI
                LOOP MIRAR_I
FI:             MOV AX,4C00H        ; (d)
```

Considereu que en la memòria hi ha les següents dades:

40030h	08h
40031h	00h
40032h	36h
	00h
	00h
	01h
40036h	03h
	02h
	56h
	89h
	13h
	18h
	23h
	26h

a) Contingut de SI,DI,DS,ES,BX,CX en el punt (a).

b) Contingut de SI,DI,DS,ES en el punt (b) quan CX té el valor 6.

c) Contingut de SI,DI,CX en el punt (c).

d) Contingut de SI,DI,CX en el punt (d) i de la zona de memòria 50100h a 50107h.

PROBLEMA 9

En el programa que es presenta a continuació es mostren parts d'unes subrutines. Responeu a les preguntes que segueixen, tenint en compte que cada subrutina fa servir una pila diferent i considerem que l'execució d'aquesta part del programa comença pel procediment PRC1

```
PRC1        PROC
            MOV [102H],SS
            MOV [100H],SP
            MOV AX,1000H
            MOV SS,AX
            MOV SP,0100H

            ;Considereu que BX=1050h en aquest punt
            PUSH BX
            CALL SUBRUT      ;Considereu que PRC1 és en el seg 7000h i
                             ;que la següent instrucció té desplaçament 150h
            MOV SS,[102H]
            MOV SP,[100H]
            RET
PRC1 ENDP
SUBRUT PROC FAR
            MOV [104H],SS
            MOV [106H],SP
            MOV AX,2000H
            MOV SS,AX
            MOV SP,0100H     ;continua a la pàgina següent
            CALL SUBRUT2     ;Considereu que SUBRUT és en el seg. 7000h i
                             ;que la següent instrucció té desplaçament 252h
```

```
            ....?....           ;(1)
            ....?....           ;(2)
            RET
SUBRUT ENDP
SUBRUT2 PROC NEAR
            MOV [108H],SS
            MOV [10AH],SP
            MOV SS,[10EH]
            MOV SP,[10CH]

            PUSH AX

            MOV [10EH],SS
            MOV [10CH],SP
            MOV SS,[108H]
            MOV SP,[10AH]
            RET
SUBRUT2 ENDP
```

10CH	6000H
10EH	6000H

a) Quines instruccions hi ha d'haver en (1) i (2) perquè es pugui retornar al procediment PRC1?

b) Contingut de la memòria entre les posicions DS:104 i DS:10F.

c) Contingut de la pila que fa servir el procediment PRC1.

d) Contingut de la pila que fa servir el procediment SUBRUT.

e) Contingut de la pila que fa servir el procediment SUBRUT2.

PROBLEMA 10

D'un programa que permet la visualització de zones de memòria a la pantalla de l'ordinador, s'ha extret el fragment que s'exposa. Aquesta part té la funció de visualitzar l'adreça (desplaçament) i el contingut d'una posició de memòria. En aquest fragment no apareix la rutina CONVERSIO, que justament és la clau del procés. Així, us demanem que feu la subrutina CONVERSIO, que fa la conversió de valor binari a una cadena ASCII que representa el valor hexadecimal d'aquest valor binari (ex: 10100111, binari, s'ha de convertir en la cadena ASCII 'A7'). S'haurà de tenir en compte el tipus de pas de paràmetres, és a dir, com dóna el programa les dades a la subrutina i com són tornats els resultats al programa, cosa que es pot deduir del programa exposat.

```
CSEG SEGMENT
            ASSUME CS:CSEG,DS:CSEG,ES:CSEG,SS,CSEG
            ORG 100H
ENTR: JMP INICI
            TAULA_CARACTERS DB '0123456789ABCDEF'
```

```
            ADREÇA DW ?
            CONTINGUT DB ?
            CADENA_A DB 4 DUP(?)          ;cadena de l'adreça
                     DB 20H               ;espai en blanc
            CADENA_C DB ?,?,0Ah,0Dh,'$'   ;cadena contingut,salt de línia i
                                          ;delimitador de visualització
INICI PROC NEAR
            MOV CX,2                      ;nombre de bytes a convertir
            LEA SI,ADRECA                 ;adreça dels bytes a convertir
            LEA DI,CADENA_A               ;adreça del resultat
            CALL CONVERSIO                ;crida a la rutina de conversió
            MOV CX,1
            LEA SI,CONTINGUT
            LEA DI,CADENA_C
            CALL CONVERSIO
            MOV DX,OFFSET CADENA_A
            MOV AH,9
            INT 21H          ;visualització de l'adreça i contingut en ASCII
INICI ENDP
CSEG ENDS
            END ENTR
```

PROBLEMA 11

A) Es vol dissenyar un sistema microprocessador amb 8086 que treballarà amb el coprocessador 8087. Quin serà el mode de funcionament del 8086 que farà més eficient el seu treball conjunt amb el coprocessador?

B) Quina és la capacitat màxima d'adreçament, tant de memòria com d'E/S, del microprocessador 8086?

C) Els registres SS, DS, CS i ES poden tenir el mateix valor? En el cas que poguessin tenir un mateix valor, quines conseqüències portaria?

D) Se sap que el 8086 es caracteritza, entre altres coses, pels accessos de 16 bits que pot fer. En el cas d'accessos a memòria, també se sap que les paraules (16 bits) poden ser guardades en qualsevol posició; però, realment totes les posicions ofereixen els mateixos avantatges a l'hora de transferir una paraula? (Explicació en un màxim de tres línies)

PROBLEMA 12

Un sistema microprocessador ha de rebre informació d'un perifèric, que consisteix en un teclat. Aquest perifèric té un buffer intern de 64 bytes i el fluxe d'informació que ha de transmetre es caracteritza per ser de grups de 64bytes consecutius com a màxim, a una velocitat de

128.000bytes/s. Quan es fa ús del teclat es suposa que s'actuarà amb una velocitat mitjana de 180 pulsacions/min.

Se sap que el perifèric disposa d'un registre intern que permet saber el seu estat (informació disponible, buffer buit, buffer ple, error,...); també disposa de senyals d'interrupció, així com d'un registre de 8 bits per guardar un codi de tipus d'interrupció, i senyals que permeten la transferència per DMA.

En un prototipus del disseny, el control del perifèric i la comunicació entre aquest i el sistema microprocessador s'ha fet per programa amb consultes periòdiques cada 0.33s, aproximadament, cridant a una rutina d'atenció cada vegada que el perifèric estava disponible.

Durant les proves d'aquest prototipus s'ha observat que les feines d'E/S relacionades amb el teclat endarrerien altres parts del funcionament del sistema. Quina solució podríeu donar a aquest problema? Raoneu la resposta en un màxim de 10 línies.

Problema 13

En el disseny d'un computador s'ha realitzat la comunicació de la impressora amb el sistema a través d'una interfície de comunicació paral.lela tal com es mostra a la figura 4. Per una banda la interfície disposa d'un registre de sortida (RS) on la CPU ha de colocar el caràcter que es vol imprimir.

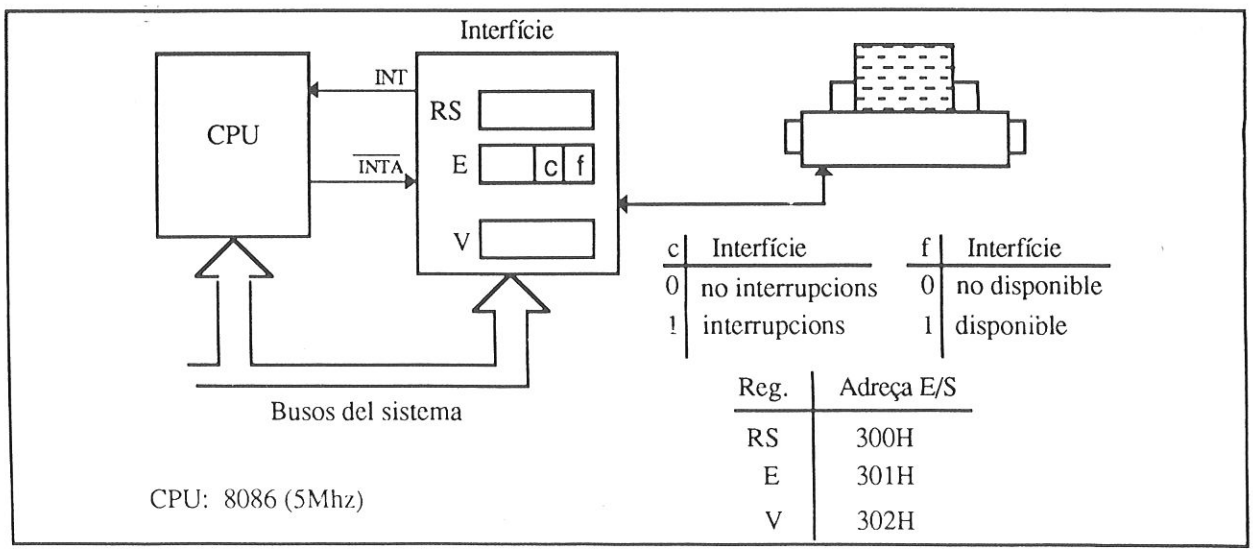


Figura 4

També disposa d'un registre de control/estat (E) que a través del seu bit 0 (f) la interfície indica (a nivell 1) quan ha enviat l'últim caràcter a la impressora i, per tant, està a punt per rebre un nou caràcter. Al mateix temps, el bit 1 d'aquest registre (c) permet controlar quan la interfície pot donar una interrupció o no, per indicar que està disponible.

Associat amb el mecanisme de les interrupcions, la interfície disposa d'un senyal de petició, INT, un senyal de reconeixement, $\overline{\text{INTA}}$, i un registre per guardar el vector d'interrupció (V) que, en aquest cas, serà el codi de vector d'interrupció, ja que la CPU és un 8086.

A la figura 4 es mostren, també, les adreces d'entrada/sortida que s'han assignat en el disseny a cadascun dels registres de la interfície.

Part A

Per realitzar la comunicació entre la impressora i la CPU s'ha optat per aplicar tècniques de consulta d'estat. Per això es disposa de la rutina d'entrada/sortida que es mostra a la figura 5, que permet transferir un caràcter a la impressora a través de la interfície. En la figura 5 també es mostra part del programa principal que farà ús de la rutina d'entrada/sortida.

A1.- Completar les instruccions (2) i (7) per tal que la rutina pugui fer la feina que li correspont.

A2.- Calcular el temps mínim d'atenció definit com el temps necessari per executar des de la instrucció (11) fins que es realitza la primera transferència de caràcter a la interfície, suposant que està disponible.

A3.- Trobar el periode de transferència definit com el temps que passa des que s'ha fet una transferència a la interfície fins que es fa la següent transferència, suposant que la interfície està disponible.

Part B

Considerant que s'opta per una segona solució en el disseny de la comunicació entre interfície i CPU que consisteix en un tractament per interrupcions, fent que la interfície demani interrupcions de tipus 60h cada cop que estigui disponible, i es realitzen modificacions en les rutines d'inicialització del sistema, tal com es mostra en la figura 6.

```
DADES     SEGMENT
  CARACTERS  DB     20 DUP(?)
          ...
DADES     ENDS
PILA      SEGMENT STACK
          DW 1000 DUP (?)
PILA      ENDS
CODI      SEGMENT
          ASSUME CS: CODI, DS: DADES, SS: PILA
ENTRADA:  MOV       AX, DADES
          MOV       DS,AX
          ...
          MOV       CX, 10                    ; (11)    => 4 cicles
          MOV       SI,0                      ; (12)    => 4 cicles
IMPR:     MOV       BL, CARACTERS[SI]         ; (13)    => 17 cicles
          CALL      RUT_IMP                   ; (14)    => 19 cicles
          INC       SI                        ; (15)    => 2 cicles
          LOOP      IMPR                      ; (16)    => 17 o 5 cicles
          ...
RUT_IMP   PROC
          PUSH      DX          ; (1)                   => 11 cicles
          MOV       DX, .......  ; (2)                  => 4 cicles
ESPERA:   IN        AL, DX      ; (3)     , AL <- (DX)  => 8 cicles
          TEST      AL,01H      ; (4)                   => 4 cicles
          JZ        ESPERA      ; (5)                   => 18 o 6 cicles
          MOV       AL, BL      ; (6)                   => 2 cicles
          MOV       DX, .......  ; (7)                  => 4 cicles
          OUT       DX, AL      ; (8)     , (DX) <- AL  => 8 cicles
          POP       DX          ; (9)                   => 8 cicles
          RET                   ; (10)                  => 8 cicles
RUT_IMP   ENDP
CODI      ENDS
          END ENTRADA
```

Figura 5

Per altra banda, el la figura 7 es mostra la rutina d'atenció a la interrupció que demanarà la interfície cada cop que estigui disponible, així com la zona de dades que es necessitarà per a l'atenció de la impressora.

```
          ...
          MOV       AL, ......    ; (A)
          MOV       DX, 302H
          OUT       DX, AL        ; inicialitza codi tipus
          PUSH      DS
          MOV       AX, 0
          MOV       DS, AX
          MOV       SI,........                 ; (B)
          MOV       [SI], OFFSET ..........     ; (C)
          MOV       [SI+2], SEG RUSIMP
          POP       DS
          MOV       AL, 00000010B     ; La interfície pot generar interrupcions
          MOV       DX, ........      ; (E)
          OUT       DX, AL            ;
          STI                         ; (D)
```

Figura 6

```
RUSIMP    PROC    FAR
          PUSH    DX                ; (17)    => 11 cicles
          PUSH    BX                ; (18)    => 11 cicles
          PUSH    AX                ; (19)    => 11 cicles
          MOV     BX, PUNT          : (20)    => 14 cicles
          MOV     AL, DATIMP[BX]    ; (21)    =>17 cicles
          MOV     DX, .......       ; (22)    => 4 cicles
          OUT     DX, AL            ; (23)    => 8 cicles
          INC     BX                ; (24)    => 2 cicles
          MOV     PUNT, BX          ; (25)    => 14 cicles
          POP     AX                ; (26)    => 8 cicles
          POP     BX                ; (27)    => 8 cicles
          POP     DX                ; (28)    => 8 cicles
          IRET                      ; (29)    => 24 cicles
RUSIMP    ENDP
          ...
DADES     SEGMENT
  DATIMP  DB 20 DUP(?)
  PUNT    DW 0
          ...
```

Figura 7

B1.- Completar les instruccions del punt (A), (B), (C), (E) i (22).

B2.- Per què serveix i com afecta als registres del 8086 la instrucció (D) de la figura 6?

B3.- Calculeu el periode de transferència (definit igual que en A3) i la velocitat de transferència màxima, tenint en compte que el temps de resposta (temps des que es produeix la interrupció fins que es comença a executar la rutina) és de 68 a 252 cicles.

B4.- Si la impressora fos capaç de tenir una velocitat d'impressió igual o major que la velocitat de transferència màxima calculada en l'apartat B3, però aquesta velocitat només la pot mantenir durant la transferència de 100 caràcters, com a màxim, quines modificacions farieu en la RUSIMP per millorar l'eficiència, pel que fa al periode de transferència, de la comunicació amb la impressora.

Problema 14

En la figura 8 es mostra un diagrama de blocs d'un sistema amb microprocessador que, entre altres perifèrics, ha de controlar una pantalla per visualitzar informació en forma de texte. Com que la comunicació amb la pantalla s'ha de fer a través d'un enllaç RS232 és necessari incorporar una interfície per adaptar la comunicació entre la CPU i la pantalla.

La interfície disposa d'un registre de dades (RD) on es guarda el caràcter a transmetre a la pantalla. També disposa d'un flag per saber si està disponible per enviar un nou caràcter (1 disponible, 0 no disponible).

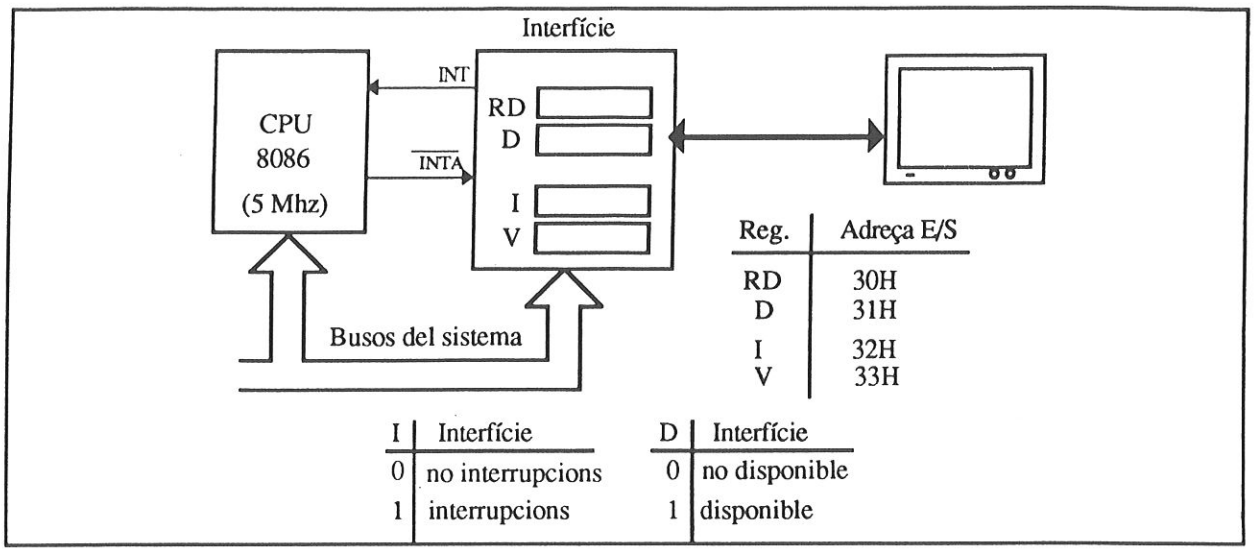


Figura 8

Per fer possible el control de la interfície a través d'interrupcions aquesta disposa de dos registres més: V que permet guardar el codi de tipus d'interrupció, i I que permet (I = 1) que la interfície generi una interrupció a través de la seva sortida INT quan s'ha transmès el caràcter que s'havia guardat en RD.

Part A

Com a primera opció per al control i comunicació amb la pantalla s'ha considerat l'aplicació de consultes d'estat.

En la figura 9 es mostra la rutina que permet transferir un caràcter a la interfície perquè siguin transmesos. També es mostra en la figura 10 part del programa principal que fa ús de la rutina d'entrada/sortida.

```
RUT_TER PROC     FAR
        PUSH     AX          ; (1)  => 11 cicles
CONSULT: IN AL, .......      ; (2)  => 10
         TEST AL, 01H        ; (3)  => 4
         JZ CONSULT          ; (4)  => 16 o 4
         MOV AL, AH          ; (5)  => 2
         OUT ......., AL     ; (6)  => 10
         ................    ; (7)  => 8
         RET                 ; (8)  => 18
RUT_TER ENDP
```

Figura 9

```
DADES    SEGMENT
         CARACTERS          DB 80 DUP(?)
         ....
DADES    ENDS
PILA     SEGMENT  STACK
         DW  100 DUP(?)
PILA     ENDS
CODI     SEGMENT
         ASSUME  CS:CODI, DS:DADES, SS:PILA
ENTRADA:          MOV AX, DADES
         MOV DS,AX
         ....
         MOV CH, 0DH                ; (9)   => 4  (#)
         MOV SI,0                   ; (10)  => 4
VISU:    MOV AH, CARACTERS[SI]      ; (11)  => 17
         CALL RUT_TER               ; (12)  => 28
         INC SI                     ; (13)  => 2
         CMP  AH, CH                ; (14)  => 3
         JNE  VISU                  ; (15)  => 16 o 4
         ....
```

Figura 10

A1) Completar les instruccions 2, 6 i 7 per tal que la rutina pugui fer la feina que li pertoca.

A2) Calcular el temps mínim d'atenció definit com el temps necessari per executar des de la instrucció (#) fins que es realitza la primera transferència de caràcter a la interfície, suposant que està disponible. Trobar el periòde de transferència: Temps entre dues transferències consecutives.

A3) Trobar el temps que es tardarà en transferir una línia de caràcters, sabent que una línia es acabada amb el caràcter de control 0Dh (CR) i que N és el número de caràcters de la línia.

Part B

Suposem, ara, que hi ha dos terminals (pantalles) que es comuniquen amb la CPU a través de dues interfícies iguals a les de la part A, una adreçada a partir del port 30h i l'altre a partir del port 40h. Tenint en compte que el programa principal s'ha modificat de la forma que es mostra a la figura 11 reponeu a les preguntes que es plantegen.

B1) Calculeu el temps mínim d'atenció (temps des de la instrucció (#) fins que es fa la primera transferència a la interfície) de cada periféric:

a) en el cas que la instrucció (30) sigui JMP ACAB1

b) en el cas que sigui NOP.

Nota: cal tenir en compte que per indicar que s'ha acabat la transferència d'una línia de caràcters a la interfície corresponent es posa el valor 80 en el registre punter (SI o DI) corresponent.

```
        ...
        MOV CH, 0DH              ; (16)   => 4 cicles  (#)
        MOV SI, 0                ; (17)   => 4
        MOV DI, SI               ; (18)   => 2
ACAB1:  CMP SI, 80               ; (19)   => 4
        JE  ACAB2                ; (20)   => 16 o 4
        IN AL, 31H               ; (21)   => 10
        TEST AL, 1               ; (22)   => 4
        JZ  ACAB2                ; (23)   => 16 o 4
        MOV AL, CARAC1[SI]       ; (24)   => 17
        INC SI                   ; (25)   => 2
        OUT 30H, AL              ; (26)   => 10
        CMP AL, CH               ; (27)   => 3
        JNE  SALT1               ; (28)   => 16 o 4
        MOV SI, 80               ; (29)   => 4
SALT1:  JMP ACAB1      (NOP)     ; (30)   => 15  (3)
ACAB2:  CMP DI, 80               ; (31)   => 4
        JE FI                    ; (32)   => 16 o 4
        IN AL, 41H               ; (33)   => 10
        TEST AL, 1               ; (34)   => 4
        JZ ACAB1                 ; (35)   => 16 o 4
        MOV AL, CARAC2[DI]       ; (36)   => 17
        INC DI                   ; (37)   => 2
        OUT 40H, AL              ; (38)   => 10
        CMP AL, CH               ; (39)   => 3
        JNE SALT2                ; (40)   => 16 o 4
        MOV DI, 80               ; (41)   => 4
SALT2:  JMP ACAB1                ; (42)   => 15
FI:     ....
```

Figura 11

Part C

Canviem la manera de controlar a la interfície fent que sigui a través de interrupcions i considerem que només hi ha una pantalla a controlar. La rutina d'atenció a la interrupció que activarà la interfície quan estigui disponible per transmetre un nou caràcter és a la posició 70012h de memòria i que el codi de tipus d'interrupció que proporciona és el 42h. A la figura 12 es detalla part de les rutines d'inicialització del sistema i en la figura 13 es dóna la rutina d'atenció a la interrupció.

```
...
MOV AL, ......          ; (A)
OUT 33H, AL
PUSH DS
XOR AX, AX
MOV DS, AX
MOV SI, .......         ; (B)
MOV [SI], OFFSET ........; (C)
MOV [SI+2], SEG ..........; (D)
POP DS
MOV AL, 01
OUT ........, AL        ; (E)
STI
...
```

Figura 12

C1) Completar les instruccions A, B, C, D i E

C2) Calcular el temps mínim d'atenció i la velocitat de transferència màxima sabent que el temps de resposta és de 68 a 252 cicles de rellotge.

```
RUT_I     PROC      FAR
          PUSH  AX                ; (43)   => 11
          PUSH  SI                ; (44)   => 11
          MOV SI, INDEX           ; (45)   => 14
          MOV AL, CARAC[SI]       ; (46)   => 17
          OUT 30H, AL             ; (47)   => 10
          INC SI                  ; (48)   => 2
          MOV INDEX, SI           ; (49)   => 15
          POP SI                  ; (50)   => 8
          POP AX                  ; (51)   => 8
          IRET                    ; (52)   => 24
RUT_I     ENDP
DADES_I   SEGMENT
          CARAC DB 80 DUP(?)
          INDEX  DB ?
          ...
DADES_I   ENDS
```

Figura 13

Problema 15

En un sistema multiprocessador (4 processadors) s'executa un programa assignant a cada processador una instrucció del programa de forma seqüencial primer a P1, després a P2, etc. Si la seqüència d'instruccions que s'executa és la que es dóna a continuació, digueu quin processador estarà més temps aturat degut a esperes de precedència. (Justifiqueu la resposta amb un graf de precedència i suposeu que cada instrucció consumeix 20 unitats de temps d'execució).

```
a := b + 2        ;S1
c := a            ;S2
c := c + b        ;S3
t := z - a        ;S4
q := r + 4        ;S5
p := q + a        ;S6
q := r + c        ;S7
```

Problema 16

En un sistema microprocessador hi ha 8 dispositius a controlar i s'utilitza la tècnica d'entrada/sortida per consulta d'estat. Sabent que l'estat dels dispositius es rep a través d'un registre d'estat que es troba en la posició d'entrada/sortida 16h i que un bit a 1 indica que el dispositiu corresponent està disponible i si està a 0 no disponible, digueu el temps d'espera fins que s'executa la rutina per al cas de la impressora, en la situació més favorable i en la situació més desfavorable.

7							0
IMPR	TECLA	VIDEO	MODE	DAC	ADC	PLOT	FD

La seqüència de consulta es fa amb la següent rutina d'atenció:

```
CONSUL:   IN  AL, 16H      ; (10)
    ROR   AL, 1            ; (2)
    JC    RUT_FD           ; (16 o 4)
    ;
    ROR   AL, 1            ; (2)
    JC    RUT_PLOT         ; (16 o 4)
    ;
    ROR   AL, 1            ; (2)
    JC    RUT_ADC          ; (16 o 4)
    ;
    ROR   AL, 1            ; (2)
    JC    RUT_DAC          ; (16 o 4)
    ;
    ROR   AL, 1            ; (2)
    JC    RUT_MODEM        ; (16 o 4)
    ;
    ROR   AL, 1            ; (2)
    JC    RUT_VIDEO        ; (16 o 4)
    ;
    ROR   AL, 1            ; (2)
    JC    RUT_TECLAT       ; (16 o 4)
    ;
    ROR   AL, 1            ; (2)
    JC    RUT_IMPR         ; (16 o 4)
    ;
    JMP   CONSUL           ; (15)
```

Entre parèntesis es dóna el nombre de cicles que ocupa l'execució de cada instrucció. El processador funciona a una freqüència de 5 MHz i cada rutina suposa 100 cicles de rellotge. Després d'atendre a un dispositiu es continua la consulta sobre el següent dispositiu.

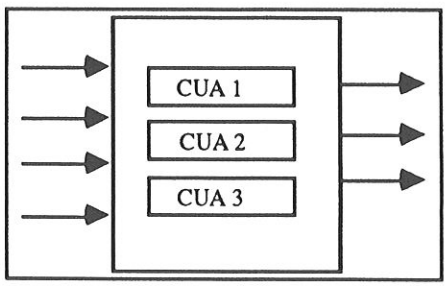
Problema 17

Un sistema de comunicacions disposa de 4 enllaços d'entrada i tres enllaços de sortida per on rep i envia informació, respectivament, en forma de paquets. El nucli de control del sistema està basat en un ordinador que executa 2 processos en paral.lel: un procés receptor i un procés emissor.

El procés receptor processa els paquets d'entrada per obtenir la destinació i els posa en la cua de paquets de la sortida que correspongui segons la destinació que indica. Cada cua té una longitud de N paquets.

El procés emissor agafa un paquet d'una cua de paquets i el transmet per la sortida corresponent.

Es demana que programeu la part del receptor i la part de l'emissor que permeten l'accés a una de les cues de paquets (per exemple a la que anomenen CUA1) de forma que s'asseguri el bon funcionament del sistema.



Problema 18

El sistema que es mostra en la figura 14 correspont a un sistema microprocessador basat en un 8086 dedicat a l'adquisició de dades i al seu posterior emmagatzemament. Bàsicament, es tracta de llegir els valors procedents d'un conjunt de conversors analògic-digitals (ADC) i guardar-los sobre una cinta magnètica en forma de blocs d'informació per fer més eficient l'accés a la unitat de cinta.

El sistema està caracteritzat per:

Sistema d'adquisició de dades

Està format per 8 conversors ADC que donen una dada de 12 bits. Per obtenir aquesta informació s'han de fer dos accessos seqüencials de byte al port d'E/S que correspongui. Cada conversor està mapejat en l'espai d'entrada/sortida com s'indica en la taula 1.

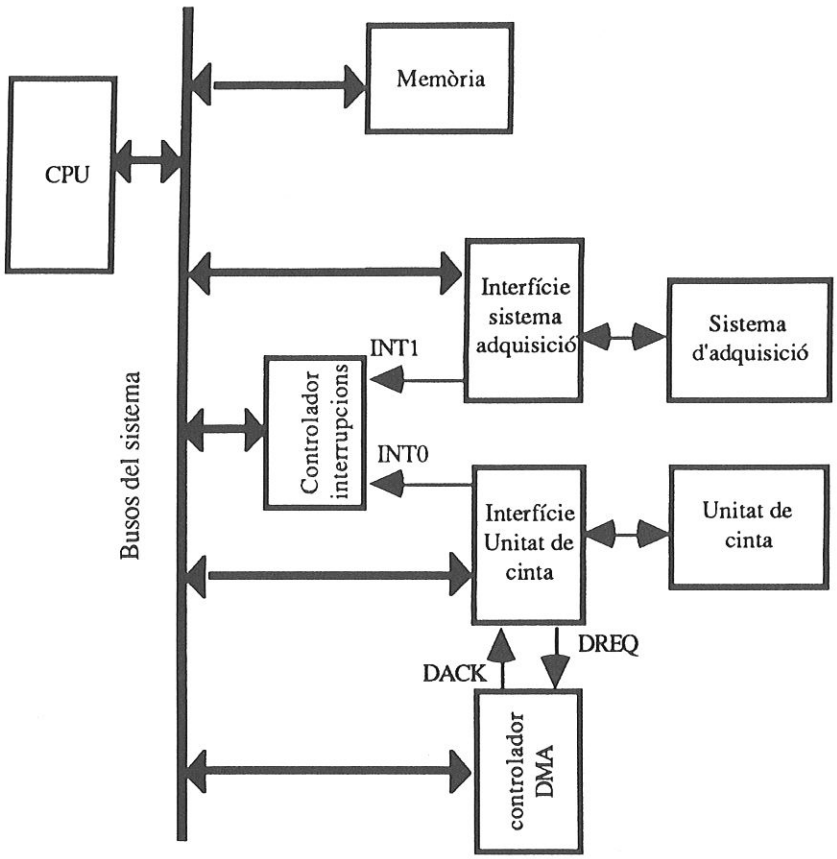


Figura 14 Diagrama de blocs del sistema del problema

Taula 4.1 Mapejat de dels conversors en l'espai d'E/S

ADC0	ADC1	ADC2	ADC3	ADC4	ADC5	ADC6	ADC7
10h	12h	14h	16h	18h	1Ah	1Ch	1Eh

Existeix un registre de control en la interfície del sistema d'adquisició, adreçat pel port 20h, del qual sabem que si el bit 0 és posat a 1 provoca l'inici d'una nova conversió en tots els ADC.

Quan tots els conversos han acabat l'última conversió demanada la interfície genera una interrupció.

L'ordre de conversió es dóna periòdicament des de una rutina de control anomenada NUCLI (figura 15). abans de donar l'ordre de conversió aquesta rutina comprova que el sistema d'unitat de cinta ha acabat de guarda la última informació que havia omplert el buffer. Això es fa a través de la variable N_BLOCS, que ha de contenir el valor 8 per indicar que es pot disposar del buffer per una nova entrada de dades.

```
M       EQU  ........              ; paràmetre per al retard del mostreig
N_BLOCS     DB      8              ; Nombre de blocs del buffer
NUCLI       PROC
INICI:      MOV CX, M
MOSTREIG:  PUSH CX                 ; periode de mostreig
      MOV CX, 0FFFFh
MOSTIN:     LOOP  MOSTIN
      POP   CX
      LOOP MOSTREIG                ;
ESPERAR:    CMP N_BLOCS, 8             ; comprovar si transfer. a cinta.
      JNE  ESPERAR
      MOV AL, 1                    ; ordenar conversió
      OUT  20H, AL
      JMP  INICI
NUCLI       ENDP
```

Figura 15 Rutina NUCLI

Sistema d'unitat de cinta

Està composat per una interfície i un controlador de DMA dedicat a les transferències entre unitat de cinta i memòria.

La interfície disposa d'un:

Registre de control (port 30h):

Bit 0: bit d'execució. A 1 provoca l'inici d'una transferència des de o sobre la unitat de cinta.

Bit 1: amb valor 1 per a entrada i a 0 per a sortida.

Registre d'estat (port 32h):

Bit 0: amb valor 1 indica unitat de cinta ocupada.

Bit 1: amb valor 1 indica un error de paritat en la transferència.

Bit 2: amb valor 1 indica que s'ha detectat el final de cinta durant la transferència.

El controlador de DMA disposa d'un:

Registre de control (port 40h):

Bit 0: amb valor 1 per fer entrades i a 0 per fer sortides.

Bit 1: amb valor 1 reconeix les peticions de DMA fetes per la interfície.

Bit 2: quan està a 1 envia una petició d'interrupció quan acaba una transferència de bloc.

Bit 3: a 1 incrementa el registre d'adreces i amb 0 el drecrementa.

Bit 4: amb 1 programa mode burst, amb 0 mode byte.

Registre d'estat (port 42h):

bit 0: amb valor 1 indica que la transferència s'ha acabat.

Registre comptador de bytes, de 16 bits, (port 44h): conté la longitud del bloc. Per inicialitzar aquest registre s'han de fer dues transferències de byte sobre el mateix port, una per donar la part baixa i una altra per donar la part alta.

Registre d'adreça, de 16 bits, (port 46h): conté el 16 bits baixos de l'adreça física de la zona de memòria que intervé en la transferència. La inicialització d'aquest registre es fa com en el registre anterior.

Port de la part alta de l'adreça (port 50h): en aquest port s'han de carregar els 4 bits alts de l'adreça física de memòria que intervé en la transferència.

Sistema de gestió d'interrupcions

Aquesta part del sistema està basada en un controlador d'interrupcions programable que té dues entrades de petició d'interrupció amb una configuració per hardware que dóna major prioritat a l'entrada INT0. El controlador disposa de tres registres interns:

Registre d'estat (port 60h):

Bit 0: estarà a valor 1 quan la petició cursada sigui la rebuda per l'entrada INT0.

Bit 1: estarà a valor 1 quan la petició cursada sigui la rebuda per l'entrada INT1.

Registre de control (port 62h):

Bit 0: si té el valor 1 permet cursar la petició feta per l'entrada INT0.

Bit 1: si té el valor 1 permet cursar la petició feta per l'entrada INT1.

Registre de vector d'interrupció (port 64h): En aquest registre es donen els 7 bits de major pes del codi de tipus de vector d'interrupció. El bit de menor pes serà posat a 0 o a 1, automàticament pel controlador d'interrupcions, segons que la interrupció cursada sigui la procedent de l'entrada INT0 o de l'entrada INT1, respectivament.

El sistema haurà de tenir dues rutines d'atenció a la interrupció, una pel sistema d'adquisició de dades (RUT_ADQ) i una altra a pel sistema d'unitat de cinta (RUT_CINTA).

A) Escriure un conjunt d'instruccions que inicialitzi la taula de vectors d'interrupció i que prepari al processador per reconèixer interrupcions, tenint en compte que la rutina d'atenció del sistema d'unitat de cinta està associada al codi de tipus 50h.

B) Escriure la rutina que prepara al controlador d'interrupcions per poder actuar correctament en el moment de rebre una petició d'interrupció.

C) Escriure la rutina d'atenció a la interrupció del sistema d'adquisició de dades. Les dades entrades es guardaran en un buffer de bytes que comença en DADES_ADQ i que té una capacitat de 2Kbytes. També controla la variable N_LECT que indica el nombre de lectures (una lectura equival a obtenir el valor de tots els ADC de l'última conversió) que s'han fet sobre el sistema d'adquisició.

Quan s'ha omplert el buffer, la rutina ha de preparar al sistema d'unitat de cinta per transferir la informació del buffer mitjantçant DMA en mode burst amb blocs de 256 bytes, de forma que es generi una interrupció al final de la transferència d'un bloc. Cal tenir en compte que abans de poder preparar una transferència sobre cinta s'ha de comprovar que la cinta no estigui ocupada. Per resoldre aquest apartat s'ha de completar la rutina que es dóna en la figura 16.

```
N_LECT          DB      0
DADES_ADQ       DB      2048 DUP(0)         ; buffer
P_DADES_ADQ     DW      OFFSET  DADES_ADQ       ; punter al buffer

RUT_ADQ     PROC
                ...
                        ; Lectura dels ADC
                ...
                        ; actualitzar la variable N_LECT
                        ; comprovar si és la última lectura (=> buffer ple)
                        ; si és ple saltar a preparar transferència a cinta
                        ; sino retornar
PREP_CINTA:     ; preparar transferència a cinta
                ...
                MOV CL,4            ; calcular adreça física de memòria per fer trans. per DMA
                MOV DX, SEG DADES_ADQ
                MOV BL, 0
MUL16:          SHL  DX, 1
                RCL  BL,1
                LOOP  MUL16
                ADD  DX, OFFSET DADES_ADQ
                ADC  BL, 0
                ...         ; inicilitzar registre d'adreces, etc.
                ...
RUT_ADQ     ENDP
```

Figura 16 Rutina RUT_ADQ

D) Escriure la rutina d'atenció a la interrupció del controlador de DMA. En aquesta rutina s'ha de comprovar si s'han produït errors en la última transferència (de paritat o de final de cinta) saltant la rutina de tractament corresponent: ERROR_PAR i ERROR_CINTA.

E) Calculeu el temps necessari per transmetre els 2kbytes del buffer a la cinta si:

La freqüència de rellotge del sistema és de 5 MHz.
Una transferència de DMA comporta 3 cicles de rellotge.
La transferència del control dels busos comporta 2 cicles de rellotge.
El retorn del control dels busos comporta 2 cicles de rellotge.
Velocitat de transferència de la unitat de cinta: 100KB/s.
El temps de resposta a una interrupció és de 68 a 252 cicles de rellotge.
El temps d'execució de les instruccions és:

MOV m, reg ; MOV reg, m :	14T
MOV reg, reg :	2T
MOV reg, valor inmediat	4T
Salt condicional	16 o 4 T
Salt incondicional	16T
LOOP	17 o 5 T
RET	8T
IRET	24T'
Rotacions i desplaçaments	4T
Operacions aritmètiques i lògiques	4T
Operacions d'E/S	8T
PUSH i POP	8T
Altres instruccions	8T

T és la durada d'un cicle de rellotge.

F) Amb el resultat del apartat E dir si seria possible mostrejar (fer lectures del sistema d'adquisició) a una freqüència de 1KHz i si no fos possible que és el que s'ha de modificar.

G) Quin valor donaries al símbol M per poder mostrejar a una freqüència de 1Hz?

Problema 19

Una base de dades bibliogràfica permet que sigui consultada per més d'una persona simultàniament. Malgrat això, degut a la capacitat de memòria i a l'estructura del sistema de consulta del programa gestor de la base de dades no es poden cursar més de 20 consultes simultànies a la base de dades. Escruire el conjunt d'instruccions que s'hauran d'incloure en el programa gestor de la base de dades

per assegurar que no sobrepassi la capacitat del sistema (o sigui, perquè no hi hagi més de 20 consultes simultànies).

Problema 20

Un sistema basat en el microprocessador 8086 ha de gestionar 3 perifèrics mitjançant interrupcions.

A) Sabent que el perifèric 1, 2 i 3 donen el codi de tipus 50h, 51h i 52h, respectivament, doneu un conjunt d'instruccions que prepari la taula de vectors d'interrupció per al tractament de d'aquests perifèrics.

B) De quin tipus de programa formarien part aquestes instruccions?

Dades:

Rutina perifèric 1 en l'adreça física 70500h
Rutina perifèric 2 en l'adreça física 08000h
Rutina perifèric 3 en l'adreça física 10004h

Problema 21

A la figura 17 es mostra part d'un sistema microprocessador on es pot veure la gestió d'una unitat de disc. La major part del temps utilitzat per la unitat de disc per accedir a la informació és per posicionar el capçal de lectura/escriptura en la pista adequada i per trobar el sector desitjat.

A partir d'aquest moment la unitat transfereix la informació a gran velocitat i el sistema ha de ser capaç de captar-la a aquesta velocitat, ja que ve donada per la velocitat de rotació del disc.

A) Si es dóna el següent grup d'instruccions per tractar la transferència a través de consultes d'estat, comprovar si és una tècnica adequada per al tractament d'una unitat de disc:

```
CONSUL:  IN AL, PORT_ESTAT_DISC   ; 10T
         TEST AL, 01H             ; 3T
         JZ CONSUL_FI             ; 4T, 16T
         IN  AL, PORT_DADES_DISC  ; 10T
         STOSB                    ; 10T
         JMP     CONSUL           ; 16T
CONSUL_FI: ...
```

B) Com a alternativa a la tècnica de consulta d'estat es proposa aplicar DMA per fer les transferències entre el disc i el sistema. Doneu un grup d'instruccions que inicialitzi el controlador DMA sabent que:

- s'ha de preparar per fer lectures en el disc,
- ha d'enviar una petició d'interrupció al final de la transferència d'un bloc de 256 bytes,
- s'han de guardar en la zona de memòria que va de 7000:100h a 7000:1FFh.

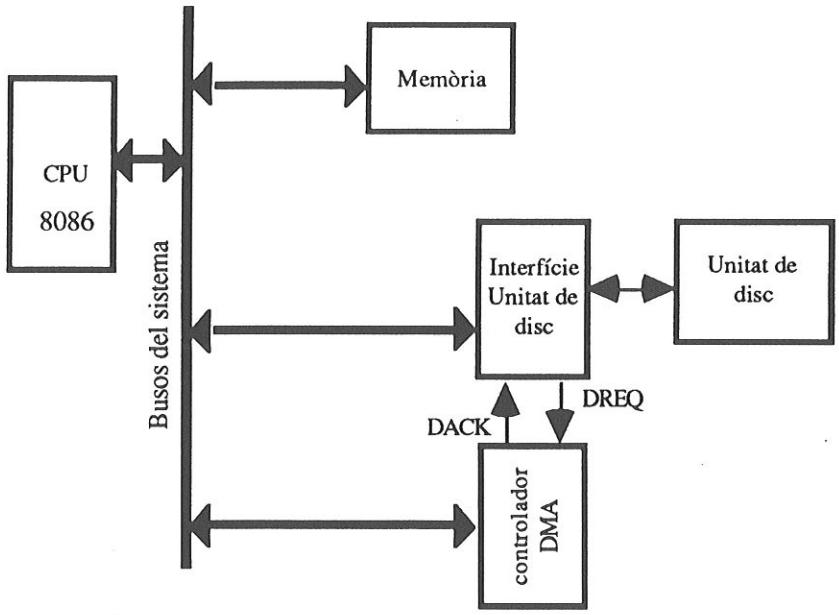


Figura 17 Diagrama de blocs del sistema

C) Calcular la velocitat de transferència que es podria aconseguir amb la tècnica d'E/S de l'apartat anterior.

Dades:

Freqüència de rellotge del sistema: 5MHz.

Una transferència de DMA comporta 3 cicles de rellotge.

La transferència del control dels busos comporta 2 cicles de rellotge.

El retorn del control dels busos comporta 2 cicles de rellotge.

Velocitat de transferència de la unitat de disc: 250KB/s.

La rutina que fa la conversió d'adreça lògica a adreça física és la següent:

```
        MOV CL,4          ; calcular adreça física de memòria per fer trans. per DMA
        MOV DX, ........  ; carregar en DX l'adreça de segment
        MOV BL, 0
MUL16:  SHL DX, 1
        RCL BL,1
```

```
LOOP  MUL16
ADD  DX, .........        ; sumar a DX l'adreça de desplaçament
ADC  BL, 0                ; en BL, DX hi haurà l'adreça física carresponent.
```

Sistema d'unitat de disc

Està composat per una interfície i un controlador de DMA dedicat a les transferències entre unitat de disc i memòria.

El controlador de DMA disposa d'un:

Registre de control (port 40h):

Bit 0: amb valor 1 per fer entrades i a 0 per fer sortides.

Bit 1: amb valor 1 reconeix les peticions de DMA fetes per la interfície.

Bit 2: quan està a 1 envia una petició d'interrupció quan acaba una transferència de bloc.

Bit 3: a 1 incrementa el registre d'adreces i amb 0 el drecrementa.

Bit 4: amb 1 programa mode burst, amb 0 mode byte.

Registre d'estat (port 42h):

bit 0: amb valor 1 indica que la transferència s'ha acabat.

Registre comptador de bytes, de 16 bits, (port 44h): conté la longitud del bloc. Per inicialitzar aquest registre s'han de fer dues transferències de byte sobre el mateix port, una per donar la part baixa i una altra per donar la part alta.

Registre d'adreça, de 16 bits, (port 46h): conté el 16 bits baixos de l'adreça física de la zona de memòria que intervé en la transferència. La inicialització d'aquest registre es fa com en el registre anterior.

Port de la part alta de l'adreça (port 50h): en aquest port s'han de carregar els 4 bits alts de l'adreça física de memòria que intervé en la transferència.

Problema 22

Un sistema d'entrada/sortida es vol gestionar per consultes d'estat. Se sap que la interfície que comunica amb el perifèric té dos registres, un de dades i un d'estat, que han estat mapejats a l'adreça

32h i 34h de l'espai d'entrada/sortida. El registre d'estat indica, amb el seu bit de menor pes, quan es pot fer una nova transferència (bit 0 a valor 1).

a) Fer una rutina que permeti controlar i realitzar la tranferència entre la CPU i perifèric, tenint en compte que es tracta d'un perifèric de sortida més lent que la CPU (RUT_SORT) i que la dada a transmetre està dipositada en la variable de byte DADA.

b) Calcular el temps mínim d'atenció, considerat a partir del començament de la rutina.

c) Calcular la velocitat de transferència màxima que es pot aconseguir amb aquesta rutina i la velocitat real de transferència si sabem que el perifèric té un temps de resposta (t_{rd}) de 400µs.

DADES:

Microprocessador: 8086 en mode màxim.

Temps d'execució: Instruccions de transferència: 8T

Salts condicionals: (4 o 16)T

Salts absoluts i salts a subrutina: 15T

Lògiques i aritmètiques: 4T

Retorn de subrutina: 18T

Freqüència de rellotge del sistema: 5MHz

Problema 23

La comunicació d'un ordinador amb una impressora es vol fer a través d'interrupcions. Es fa servir una interfície per adaptar aquesta comunicació de la qual sabem que disposa d'un registre de dades, un registre d'estat, un registre de control i un registre de vector d'interrupció.

El bit 0 del registre d'estat es posa a 1 quan la interfície pot acceptar un nou byte per ser transmès a la impressora.

Quan el bit 0 del registre de control és inicialitzat a 1 la interfície genera una petició d'interrupció, amb codi de tipus 68h, per indicar que pot acceptar un nou byte. El senyal de petició d'interrupció de la interfície està connectat a l'entrada INTR d'un 8086.

La interfície ha estat situada a partir de l'adreça 52h i ocupa posicions amb adreça parella del mapa d'entrada/sortida, que permeten accedir al seu registre de dades, estat, control i de vector d'interrupció, respectivament.

La rutina de servei a la interrupció s'anomena RUT_SER.

A) Escriure la rutina d'inicialització del sistema que faci possible el tractament correcte de les interrupcions demanades a la CPU.

B) Escriure la rutina de servei RUT_SER tenint en compte que les dades a imprimir es troben en una variable buffer (DADES_I) amb capacitat per a 100 bytes, tant gran com el buffer de dades que té internament la impressora. En la variable PUNT_D es considera que hi ha l'adreça del següent byte a transmetre.

C) Calcular la velocitat de transferència màxima que permet la rutina dissenyada en l'apartat B.

DADES:

Freqüència de rellotge del sistema: 5MHz
Microprocessador: 8086 en mode màxim.
Temps d'execució: Instruccions de transferència: 8T
Salts condicionals: (4 o 16)T
Salts absoluts i salts a subrutina: 15T
Lògiques i aritmètiques: 4T
Retorn de subrutina d'interrupció: 24T
Temps de resposta a les interrupcions: (68 a 252)T
Temps de resposta del dispositiu (t_{rd}): 20μs

Problema 24

En la figura 24.1 es mostra part d'un sistema microprocessador on es pot veure la gestió d'una unitat de disc. La major part del temps utilitzat per la unitat de disc per accedir a la informació és per posicionar el capçal de lectura/escriptura en la pista adequada i per trobar el sector desitjat. A partir d'aquest moment la unitat transfereix la informació a gran velocitat i el sistema ha de ser capaç de captar-la a aquesta velocitat, ja que ve donada per la velocitat de rotació del disc.

A) Per poder atendre les necessitats del sistema de disc es proposa aplicar DMA per fer les transferències entre el disc i el sistema. Doneu un grup d'instruccions que inicialitzi el controlador DMA sabent que:

- s'ha de preparar per fer escriptures en el disc,
- ha d'enviar una petició d'interrupció al final de la transferència d'un bloc de 512 bytes,

- s'han de transferir les dades de la zona de memòria que va de 4400:0000h a 4400:01FFh.

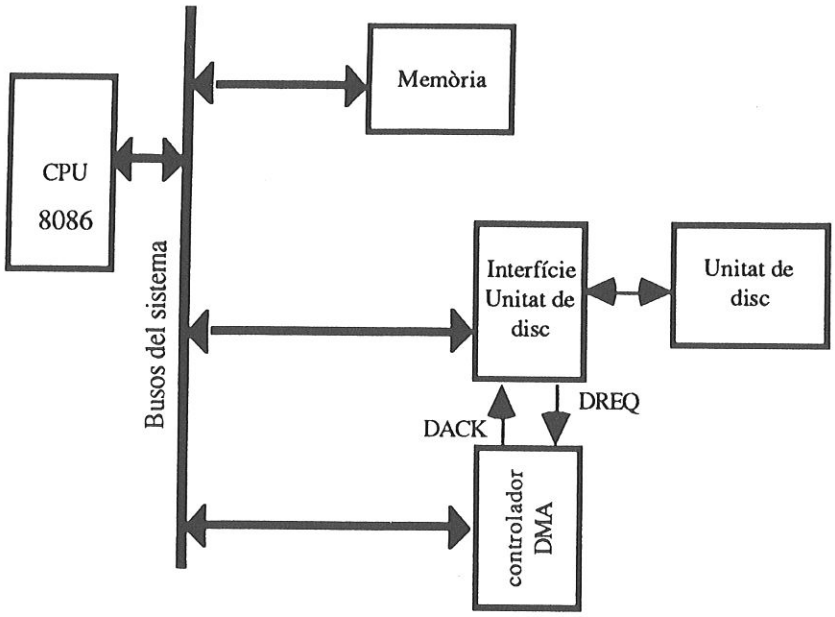


Figura 24.1. Diagrama de blocs del sistema del problema 24.

B) Calcular la velocitat de transferència que es podria aconseguir amb la tècnica d'E/S de l'apartat anterior.

Dades:

Freqüència de rellotge del sistema: 5MHz

Una transferència de DMA comporta 4 cicles de rellotge.

La transferència del control dels busos comporta 2 cicles de rellotge.

El retorn del control dels busos comporta 2 cicles de rellotge.

Velocitat de transferència de la unitat de disc: 250KB/s.

Sistema d'unitat de disc

Està composat per una interfície i un controlador de DMA dedicat a les transferències entre unitat de disc i memòria. El controlador de DMA disposa d'un:

Registre de control (port 60h):

Bit 0: amb valor 1 reconeix les peticions de DMA fetes per la interfície.

Bit 1: amb valor 1 per fer entrades i a 0 per fer sortides.

Bit 2: quan està a 1 envia una petició d'interrupció quan acaba una transferència de bloc.

Bit 3: a 1 incrementa el registre d'adreces i amb 0 el drecrementa.

Bit 4: amb 1 programa mode burst, amb 0 mode byte.

Registre d'estat (port 62h):

bit 0: amb valor 1 indica que la transferència s'ha acabat.

Registre comptador de bytes, de 16 bits, (port 64h): conté la longitud del bloc. Per inicialitzar aquest registre s'han de fer dues transferències de byte seguides sobre el mateix port, una per donar la part baixa i una altra per donar la part alta.

Registre d'adreça, de 16 bits, (port 66h): conté el 16 bits baixos de l'adreça física de la zona de memòria que intervé en la transferència. La inicialització d'aquest registre es fa com en el registre anterior.

Port de la part alta de l'adreça (port 70h): en aquest port s'han de carregar els 4 bits alts de l'adreça física de memòria que intervé en la transferència DMA.

Problema 25

Un sistema basat en un microprocessador 8086 en mode mínim rep interrupcions procedents d'un controlador d'interrupcions i també té un subsistema de DMA per a la comunicació amb perifèrics ràpids. Ens interessa estudiar el correcte funcionament del sistema sabent que es poden produir interrupions i peticions de DMA en qualsevol moment.

A) Si, en un principi, el subsistema DMA fa transferències de bloc (en mode burst) de 512 bytes, pot sorgir algun problema d'atenció a la resta del perifèrics durant la transferència per DMA?

B) Tenint en compte les necessitats d'atenció d'una interrupció demanada per un perifèric, quina ha de ser la longitud òptima de bloc per a una transferència per DMA que compleixen aquestes necessitats?

DADES:

Freqüència de rellotge del sistema: 5 MHz

Velocitat de transferència per DMA: 1 MB/s

Temps per transferir el control del busos: 3 μs

Temps màxim d'espera de reconeixement a una petició d'interrupció: 100 μs

Temps de resposta a una interrupció (t_{resp}): 68T (Considerant menyspreable el temps per acabar una instrucció).

Temps mínim d'atenció de la rutina d'interrucpció (t_{ma}): 20 µs

Problema 26

La comunicació amb una impressora es vol controlar a través de consultes d'estat. La interfície corresponent disposa d'un registre de dades i un d'estat. Suposant que el programa que controla el procés d'E/S de la impressora és el que es dóna seguidament responeu a les qüestions plantejades.

```
RUT_IMP   PROC
          PUSH AX
SALT1:    IN AL, 20H
          TEST AL, 1
          JZ SALT1
          MOV AL, VALOR
          OUT 22H, AL
          POP AX
          RET
RUT_IMP   ENDP
```

a) Quines són les adreces del port de dades i d'estat de la interfície d'impressora? La consulta d'estat es fa per bloqueig de programa o per consultes periòdiques?

b) Quantes dades es transfereixen a la impressora cada cop que es crida la rutina?

c) Quina velocitat de transferència màxima es pot aconseguir amb aquesta rutina suposant que la impressora té un t_{rd} (temps de resposta de dispositiu) prou petit?

d) Quina és la velocitat de transferència que es pot aconseguir si la impressora no té buffer de dades d'entrada i imprimeix a una velocitat de 200 cps?

DADES:

Microprocessador: 8086 en mode màxim.

Temps d'execució: Instruccions de transferència: 8T

Salts condicionals: (4 o 16)T

Salts absoluts i salts a subrutina: 15T

Lògiques i aritmètiques: 4T

Retorn de subrutina: 18T

Freqüència de rellotge del sistema: 5MHz

Problema 27

Un sistema disposa d'un teclat com a perifèric d'entrada de dades, que es comunica a través d'una interfície sèrie. La interfície que fa possible aquesta comunicació disposa d'un registre de dades, un registre d'estat i un registre de control. Les adreces d'aquests registres són 40h, 42h i 44h, respectivament. Si el bit 0 del registre d'estat és a 1 indica que hi ha una nova dada disponible. Si el bit 0 del registre de control ha estat posat a 1 la interfície genera una petició d'interrupció quan ha rebut un nou byte.

Els bits 7 i 6 del registre de control (els de major pes) tenen sempre el valor 1, de manera que els 8 bits d'aquest registre s'utilitzen com a codi de tipus d'interrupció.

A) Escriure la rutina que inicialitzi el sistema perquè respongui correctament a les peticions d'interrupció demanades per la interfície sèrie. Suposar que la rutina d'atenció es troba en la posició 4020:0100 h.

Suposant que la rutina d'atenció és la que es dóna a continuació:

```
RUT_ATEN  PROC
          PUSH AX
          PUSH DI
          MOV DI, PUNTER
          IN AL, 40H
          MOV [DI], AL
          INC DI
          MOV PUNTER, DI
          POP DI
          POP AX
          IRET
RUT_ATEN  ENDP
```

B) On es guarden les dades procedents del teclat i quina és la funció de la variable PUNTER?

C) Calcular la velocitat de transferència màxima suposant que el dispositiu està disponible.

D) Quin tant per cent del temps la CPU està ocupada en entrades per teclat (suposant "règim permanent")? Quina millora es podria donar per augmentar la velocitat de transferència i disminuir el temps dedicat a aquestes entrades?

DADES:

Freqüència de rellotge del sistema: 5MHz

Microprocessador: 8086 en mode màxim.

Temps d'execució: Instruccions de transferència: 8T

Salts condicionals: (4 o 16)T

Salts absoluts i salts a subrutina: 15T

Lògiques i aritmètiques: 4T

Retorn de subrutina d'interrupció: 24T

Temps de resposta a les interrupcions: (68 a 252)T

Velocitat mitjana de tecleig: 300 ppm.

Problema 28

Una unitat d'entrada/sortida que s'encarrega de controlar la transferència des d'una unitat de disc flexible a través de DMA, ha d'estar dissenyada per aconseguir una màxima velocitat de transferència amb una mínima desocupació de la CPU.

A) Quin mode de transferència escullirieu per complir aquestes condicions?

B) Quina capacitat té aquest disc?

Dades:

Freqüència de rellotge del sistema: 5MHz

Una transferència de DMA comporta 25 cicles de rellotge.

La transferència del control dels busos comporta 15 cicles de rellotge.

El retorn del control dels busos comporta 15 cicles de rellotge.

Unitat de disc:

- Dues cares.
- Velocitat de rotació del disc: 360 rpm.
- 80 pistes per cara.
- Densitat d'informació: 15000 bits/pista

Longitud mitjana dels blocs a transferir: 2KB

TESTS

Resposta correcta: 1 punt
Resposta incorrecta: - 1/3 punts
Resposta no contestada: 0 punts

Codi: **34013100**

Durada: 1 hora

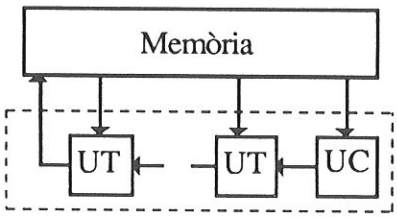
1.- Quan un microprocessador disposa d'una unitat de control microprogramada, és cert que:

x a) Per executar una instrucció, el microprocessador executa un conjunt d'instruccions més senzilles i bàsiques.
b) La unitar de control té un processador aritmètic que treballa en paral.lel amb la resta del microprocessador.
c) La unitat de control estarà constituida per un hardware molt complex.
d) Les afirmacions anterios són totes falses.

2.- Una màquina programable amb una gran varietat d'instruccions rep el nom de:

a) RISC x b) CISC c) MIMD d) SIMD

3.- L'estructura que es mostra a continuació correspon a una arquitectura:



= SIMD

a) MIMD b) RIMD c) PIMD x d) Cap de les anteriors

4.- Quin dels següents tipus de memòria es pot utilitzar per aplicar gestió de memòria virtual per paginació:

a) RAM i ROM b) ROM i memòria caché
x c) Disc i RAM d) Totes les anteriors

5.- Les principals diferències entre la paginació i la segmentació són:

a) En la paginació les pàgines es poden solapar unes amb les altres i cadascuna té una ~~longitud diferent~~; en canvi, els segments són tots de longitud diferent però no es poden solapar.
x b) En la segmentació els segments tenen longitud variable i es poden solapar mentre que en la paginació les pàgines són de longitud fix i no es poden solapar.
c) Les pàgines no poden ser majors de 64 kbytes.
d) són correctes la a) i la c)

6.- Quin dels següents microprocessadors presenta una estructura de busos multiplexats:

a) 80286 b) 80386 c) 68000 d) Cap dels anteriors

7.- Quins senyals s'encarreguen de la gestió de bus en el 8086?

a) HOLD i HLDA en mode màxim, i $\overline{RQ}/\overline{GT0}$ i $\overline{RQ}/\overline{GT1}$ en mode mínim
b) $\overline{RQ}/\overline{GT0}$
c) $\overline{RQ}/\overline{GT1}$
d) HOLD i HLDA en mode mínim, i $\overline{RQ}/\overline{GT0}$ i $\overline{RQ}/\overline{GT1}$ en mode màxim

8.- Com es calcula l'adreça física en el 8086?

a) adreça física = (16 x desplaçament) + registre de segment
b) adreça física = desplaçament + (16 x registre de segment)
c) adreça física = registre de dades + (16 x desplaçament)
d) adreça física = (16 x Registre general) + desplaçament

9.- Quines són les línies d'interrupció pel 68000?

a) FC0 - FC2 b) IPL0 - IPL2 c) INT i NMI d) BR, BG i BGACK

10.- L'espai d'adreces físic pel 68000 és:

a) 8Mwords b) 1Mbyte c) 16Mwords d) 20Mbytes

11.- Quin dels següents microprocessadors permet guardar paraules en adreces parells?

a) 8086 b) 68000 c) 80286 d) Tots els anteriors

12.- EL 8086 té dos modes de funcionament, el mode mínim i el mode màxim:

a) En mode màxim pot adreçar més memòria ja que té més terminals de control.
b) En mode màxim té capacitat per detectar errors en el sistema, com errors de paritat en els busos.
c) A nivell de software, és indeferent el mode en que es tigui.
d) En mode mínim el xip és més reduït i disposa de menys terminals i, per tant, permet construir sistemes més compactes.

13.- Què s'entén per bus assíncron?

a) Aquell en que el cicle de bus no s'acaba mentre no es rebi un senyal de reconeixement ("ACK").
b) Aquell que té un temps de transferència fixe per defecte, en el qual el dispositiu adreçat ha de respondre sense que la CPU esperi senyal de reconeixement per continuar.
c) Aquell que permet mantenir la compatibilitat amb els microprocessadors de 8 bits.
d) Cap de les anteriors

14.- El 68000 té el bus d'adreces, dades i els registres de :

a) 16, 20, 16 bits, respectivament
b) 24, 16, 16 bits, respectivament
c) 32, 16, 32 bits, respectivament
d) 24, 16, 32 bits, respectivament

15.- El 80286 és un microprocessador de:

a) 16 bits b) 32 bits c) 64 bits d) 8 bits

16.- Una característica important del 80286 és que:

a) Té una arquitectura pipeline de 5 unitats que treballen de forma seqüencial.
b) La CPU va acompanyada per una MMU.(unitat de gestió de memòria), integrada en el mateix xip.
c) Té una capacitat d'adreçament de 16Mwords.
d) És un microprocessador de 32 bits compatible amb el 8086.

17.- Quan es treballa en el 80286 en el mode real tenim que:

a) Disposa d'una capacitat de processament d'instruccions tal que permet que sigui utilitzat en qualsevol sistema en temps real.
b) Les adreces del programa són les adreces físiques.
c) S'utilitzen unes taules de mapa de memòria que ens transformen les adreces virtuals en adreces físiques.
d) A nivell de software, el 80286 funciona igual que el 8086 però amb un conjunt d'instruccions més ampli.

18.- El microprocessador 68020 per indicar el tipus d'accés a realitzar (byte, paraula o doble paraula) treu informació a través dels terminals:

a) $\overline{AS}$, $\overline{DS}$ b) SIZ0, SIZ1 c) $\overline{DSACK0}$, $\overline{DSACK1}$ d) DB, DW, DD

19.- La memòria caché del 68020 està basada en dos blocs funcionals, el buffer de tag i el buffer de dades, que tenen assignats les següents tasques:

a) Permetre augmentar la capacitat de carregabilitat (fan-out) del bus del sistema (adreces, dades i control).
b) El buffer de dades només conté els operands de les instruccions que s'han d'executar tot seguit.
c) El buffer de tag conté informació sobre l'adreça de les instruccions que conté la memòria caché.
d) Contenir les taules de traducció d'adreces virtuals a adreces físiques.

20.- Les tècniques de gestió de memòria que pot aplicar el 80386 són:

a) Al ser compatible amb el 8086 tindrà la mateixa capacitat que aquest i només aplicarà segmentació.
b) Només pot aplicar la paginació.
c) Pot combinar les tècniques de paginació i segmentació.
d) A través de la seva MMU mantindrà un sistema basat en la tècnica de registre de base i límit.

21.- Quina part del 8086 és l'encarregada de generar l'adreça física que s'ha de donar sobre el bus d'adreces?

a) la Unitat d'Execució
b) La unitat d'Interfície amb el Bus
c) La Unitat de Control d'Execució
d) La ALU (Unitat Linkadora d'Adreces)

22.- L'aplicació de la técnica de gestió de memòria per segmentació en el 8086 està limitada perquè:

a) La longitud dels segments és de 64 bytes i comencen sempre en una adreça física múltiple de 16.
b) La longitud dels segments és de 64 Kbytes i comencen sempre en l'adreça 16.
c) La longitud dels segments és de 16 Kbytes i comencen en una adreça parell.
d) la longitud dels segments és de 64 Kbytes i comencen en una adreça física múltiple de 16.

23.- La forma en que el 8086 emmagatzema una paraula a la memòria es pot representar com:

a)

Byte alt	Byte baix

en adreça m

Byte alt	m + 1
Byte baix	m

b)

Byte alt	Byte baix

en adreça n

Byte baix	n + 1
Byte alt	n

c)

Byte alt	Byte baix

en adreça n + 1

Byte alt	n + 2
Byte baix	n + 1

d) Són certes la a) i la c)

24.- En el mode d'adreçament de memòria de dades relatiu a base del 8086, l'adreça efectiva es calcula a partir de:

a) EA = ⟨SI o DI⟩ + despl
b) EA = ⟨SI o DI⟩ + ⟨BX o BP⟩ + despl
c) EA = ⟨BX o BP⟩ + despl
d) EA = ⟨BX o BP⟩ + DX

25.- Donada la instrucció MOV AL,03 el mode d'adreçament de l'operand font és:

a) directe
b) relatiu a base
c) immediat
d) Cap dels anteriors

EA= desplaçament

26.- Calcular l'adreça efectiva per al mode d'adreçament directe amb les següents dades:

BX = 0010h SI = 10A3h Despl = 1001h DS = 2001h

a) EA = 1001h b) EA = 10B3h c) EA = 20020h d) Cap de les anteriors

27.- Calcular l'adreça efectiva i física pel mode d'adreçament indexat donades les següents dades:

DS = 2100h SI = 0011h BP = 0102h despl = 0100h BX = 2000h

a) EA = 010Ah , af = 21111h
b) EA = 0110h , af = 20111h
c) EA = 0111h , af = 22111h
d) EA = 0111h , af = 21111h EA= (SI o DI) + desplo AF= DS x 10h + desplo + SI

28.- Donades les següents instruccions:

1) MOV AX,AL 2) ADD [1000H], [SI] 3) MOV DI, 3
4) PUSH ~~BH~~ 5) MOV [CX], AX 6) MUL CL

a) Són correctes la 1, 2 i 3
b) Són correctes la 3, 5 i 6
c) Només és correcta la 4
d) Són correctes la 3 i la 6

29.- El microprocessador 80386 té un mode de funcionament anomenat mode protegit. En aquest mode, el 80386:

a) No és afectat per cap tipus d'interferència electromagnètica.
b) Té més prioritat que la resta de dispositius del sistema a l'hora de controlar els busos i de demanar interrupcions.
c) Té operativa la unitat de gestió de memòria disposant de característiques per al tractament de memòria virtual , protecció i multitasca.
d) NO acceptarà cap petició d'interrupció mascarable durant l'execució de qualsevol instrucció de transferència.

30.- El mode usuari del 68000:

a) Permet utilitzar al 68000 en sistemes multiusuari.
b) No permet utilitzar la unitat de gestió de memòria.
c) Dóna facilitats addicionals al programador per poder utilitzar el microprocessador de forma més eficient.
d) Restringeix el nombre d'instruccions del 68000 que es poden executar.

Resposta correcta: 1 punt
Resposta incorrecta: - 1/3 punts
Resposta no contestada: 0 punts

Codi: 34013200

Durada: 40 minuts

1.- Per poder aconseguir que el procés de depuració del disseny d'un microprocessador fos menys costós i més flexible, la unitat de control fariem que fos construida amb:

↳ MICROPROGRAMACIÓ
↳ CABLEJADA.

a) lògica "cablejada".
b) tècniques RISC.
c) tècniques ECL.
d) tècniques de microprogramació.

2.- L'estructura d'un processador pot està formada per una matriu d'unitats de tractament on cada fila és controlada per una unitat de control diferent. En aquest cas estem parlan d'un processador:

a) Microprogramat.
b) Segmentat.
c) MIMD.
d) RISC

3.- Ens interessa dissenyar un processador amb un conjunt d'instruccions que sigui de molt ràpida execució, encara que això ens obligui a disposar de poques instruccions pel processador. Aplicarem el concepte:

a) Temps d'accés mínim a la instrucció.
b) Minimització de la lògica interna (ILM).
c) CISC.
d) Cap de les anteriors

RISC

4.- El 8086 és un microprorcessador de 16 bits que:

a) Té un bus d'adreces de 20 bits multiplexat amb el bus de dades.
b) Té els busos de dades, adreces i de control multiplexats.
c) Té un mode de funcionament per a gestió de memòria virtual anomenat mode màxim.
d) Els seus registres interns són de 32 bits.

5.- La memòria virtual en un computador s'aconsegueix:

a) Preveient futures ampliacions de memòria en la fase de disseny.
b) Dotant al sistema d'un controlador de memòria caché.
c) Escollint una CPU que permeti separar el mapa de memòria del mapa d'E/S.
d) Cap de les anteriors.

6.- La gestió dels busos en un sistema amb 68000 es fa a través dels terminals:

a) FC0-FC2
b) IPL1-IPL2
c) $\overline{RQ/GT0}$, $\overline{RQ/GT1}$
d) $\overline{BR}$, $\overline{BG}$, $\overline{BGACK}$

7.- Si en el bus d'adreces d'un 8086 apareix el valor ABCDEh, vol dir que l'adreça lògica a la que correspon és:

a) Adreça desplaçament = BCDEh Adreça segment = A000h
b) Adreça segment = ABCDh Adreça desplaçament = 000Eh

c) Adreça desplaçament = ACDEh Adreça segment = A100h
d) Totes les anteriors són vàlides

8.- Una de les característiques notables del 68000 és:

a) Disposar d'un conjunt de registres molt homogeni de 32 bits.
b) Permetre un tractament separat dels accessos a memòria i a E/S a través de mapes d'adreçament separats.
c) Disposar de totes les línies dels busos dissenyades especialment per fer transferències síncrones.
d) Cap de les anteriors.

9.- L'adreçament de paraules en posicions parells està permès en:

a) En el 68000 i el 68020.
b) Tota la família de microprocessadors de 16 bits de INTEL.
c) En el 80386DX.
d) Totes són correctes.

10.- En el tractament de les interrupcions en el 68000:

a) Es disposa de 7 nivells de prioritat més un estat de no petició.
b) Es disposa de 3 línies d'interrupció que permeten diferenciar 3 tipus d'interrupció.
c) Es disposa de 3 nivells de prioritat.
d) Cap de les anteriors és correcte.

11.- Si voguèssim dissenyar un sistema amb microprocessador amb el mínim nombre de lògica addicional i el mínim nombre de pins del xip, escolliríem: GO MINS → MULTIPLEXATS

a) un 8086 b) un 68000 en mode usuari
c) un 8086 en mode mínim d) un 80286 en mode real
SENYALS DE CONTROL GRÀFICES

12.- Quan un 80286 treballa en mode protegit:

a) Únicament quedarà afectat per la limitació en el nombre d'instruccions que es poden executar, per raons de protecció.
b) Tindrem moltes facilitats per aplicar multitasca.
c) És compatible amb el mode supervisor del 68000.
d) Cap virus podrà afectar l'execució dels programes.

13.- La capacitat d'adreçament del 68020 és:

a) 4 Gigabytes físics. b) 4 Gigabytes físics i 64 TeraBytes virtuals.
c) 1 MByte. d) 16 MBytes.

14.- Algunes diferències entre el 68000 i el 68020 són que:

a) El 68020 té un conjunt d'instruccions de rotació i desplaçament més ràpides i la taula de vectors d'interrupció no té una posició única.
b) El 68000 té una taula de vectors d'interrupció en la posició 0 i té memória caché integrada.
c) El 68020 té una MMU integrada i la taula de vectors d'interrupció no té una posició única.
d) Cap de les anteriors.

15.- L'aplicació de la técnica pipeline en el disseny de processadors:

a) Aconsegueix disminuir els costos de fabricació ja que minimitza el nombre de components.
b) Dóna característiques al processador que li permetran gestionar memòria virtual.
c) Augmenta la velocitat de processament d'intruccions.
d) Proporciona un tractament en paral.lel de la informació similar a les màquines SIMD.

16.- Quan en un computador es vol incloure memòria caché:

a) El computador haurà de disposar d'un controlador d'aquest tipus de memòria, qui la omplirà amb les instruccions físicament pròximes a la actualment executada.
b) Augmenta la seva capacitat d'adreçament de memòria.
c) Augmenta la capacitat de magatzemament de dades.
d) La memòria caché es comportarà com una extensió de la memòria virtual que permet fer més ràpida l'execució d'instruccions.

17.- La forma en que el 68000 emmagatzema una paraula a la memòria es pot representar com:

a) | Byte alt | Byte baix | en adreça m

Byte alt	m + 1
Byte baix	m

b) | Byte alt | Byte baix | en adreça n

Byte baix	n + 1
Byte alt	n

c) | Byte alt | Byte baix | en adreça n + 1

Byte alt	n + 2
Byte baix	n + 1

d) Són certes la b) i la c)

18.- Si en una instrucció de 8086 s'aplica adreçament relatiu a base, quina serà l'adreça física generada si s'accedeix al segment de dades sabent que l'estat del processador és el següent:

DX = 7002h BP = 100h BX = 077h CX = 6000h AX = 9999h SS = 9000h
ES = 7000h DS = 8000 desplaçament = 02h

a) Adreça física = 90102h
b) Adreça física = 90079h
c) Adreça física = 87004h
d) Adreça física = 80079h 80102h

AF = DS × 10h + EA.

EA = (BX o BP) + despl. EA = 79h

19.- Per a les següents instruccions:

1) ADD DX, 87H
2) POP [CX]
3) MUL BX, CL
4) MOV [BX][SI][2],AX
5) SAL AX,4

AL
DX

a) són incorrectes la 2, 3 i 5
b) són incorrectes la 1, 2, 3 i 5.
c) Són incorrectes la 1, 2 i 4
d) Són incorrectes la 1 i la 4

20.- En el mode usuari del 68000, el 68000:

a) No fa ús de la unitat de gestió de memòria virtual.
b) Només atendrà les excepcions internes.
c) Fa servir el registre A7'.
d) Cap de les anteriors.

Resposta correcta: 1 punt
Resposta incorrecta: - 1/3 punts
Resposta no contestada: 0 punts

Codi de la prova: 34013300

durada: 40 minuts

1.- En el disseny d'una CPU es vol aconseguir una unitat de control amb hardware reduït i que faciliti les tasques de depuració i millora del disseny:

A) Aplicarem tècniques de disseny top-down.
B) Escollirem una arquitectura microprogramada.
C) Farem que el processador sigui construït amb tècniques RISC.
D) Cap de les anteriors és certa.

2.- El microprocessador 8086 incorpora un sistema de gestió de memòria:

A) Per segmentació amb segments de qualsevol longitud.
B) Paginació amb tamany de pàgina de 64K.
C) Segmentació paginada ja que els segments han de començar en adreces múltiples de 16.
D) Cap de les anteriors.

3.- Les màquines pipeline permeten augmentar la velocitat de procés basant-se en:

A) Dividir a la unitat d'execució en vàries unitats de tractament que treballen en paral.lel.
B) Tenir un sistema operatiu amb multiprogramació.
C) El solapament en el temps de les diferents funcions del cicle de recerca-descodificació-execució.
D) Són correctes la A i la C.

4.- Un dels objectius dels processadors amb un conjunt d'instruccions complexe és:

A) Aconseguir que el software de nivell baix s'assembli al software d'alt nivell per obtenir una gran potència en el llenguatge màquina.
B) Obtenir processadors senzills complicant els compiladors de llenguatge d'alt nivell.
C) Obtenir un processador de propòsit general.
D) Són correctes la A i la B.

5.- Si el 8086 en els seus registres té els valors que es donen a continuació: SI = 100h, DI = 200h, AX = 0, CS = 8000h, BX = 400h, IP = 2h, DS = 7002h, ES = 8000h

A) La següent instrucció que serà executada és la que està en la posició física 80100h
B) La instrucció MOV [DI], BL posa a 0 la posició física 70220h.
C) La instrucció MOV ES:[DI], AL no és correcta ja que intenta escriure sobre el segment de codi.
D) Podriem fer un accés a la posició 400:200h utilitzant adreçament relatiu a base.

DS:DI ← BL

DS × 10h + DI

7002h × 10h + 200h

70220h Adreça física.

BX = 0400h

→ BH = 04h
BL = 00h

6.- S'està analitzant un sistema basat en un 8086 mitjançant un analitzador lògic TEKTRONIX PRISM 3001, dotat d'una sonda i un software específic per a l'anàlisi d'aquest tipus de sistemes microprocessador. És cert que:

A) Podrem comprovar el resultat d'una instrucció del tipus ADD AL,2.
B) Podrem calcular el temps d'execució d'una instrucció del tipus ADD AL,2.
C) Podrem veure com queden afectats els flags després de l'execució d'una instrucció del tipus ADD AL, 2.
D) Podrem dir quina instrucció s'ha executat abans de ADD AL,2.

7.- En un sistema basat en un sistema operatiu multiprogramat: (pag. 188)

A) Només és pòssible operar en OFF-LINE (o amb processos Batch).
B) El sistema de gestió de memòria serà senzill.
C) L'objectiu principal a l'hora de dissenyar el sistema operatiu és que el planificador de la CPU sigui ràpid.
D) Cap de les anteriors és correcte.

8.- En un processador matricial és cert que: MIMD

A) Hi ha una única unitat de control.
B) Executa en paral.lel vàries instruccions.
C) La seva unitat de control està dividida en forma matricial.
D) Cap de les anteriors.

9.- Si dos processos compleixen la condició de Bernstein podem assegurar que:

A) Els processos no són disjunts.
B) No podrem executat els dos processos en concurrència real, però sí en concurrència virtual.
C) Aquests dos processos no són cooperants.
D) La A i la C són certes.

10.- Una instrucció del 8086 escriu una paraula de valor 7204h en la posició física 0F1050h:

A) Llegint una paraula en la posició física 0F1051h obtenim una paraula amb la part baixa igual a 72h.
B) Llegint una paraula en la posició física 0F104Fh obtenim una paraula amb la part baixa igual a 04h.
C) Llegint un byte en la posició física 0F104Fh obtenim el valor 72h.
D) No podem escriure en aquesta zona de memòria perquè és una zona dedicada a memòria ROM.

11.- Un ordinador, on s'ha aplicat un sistema de gestió de memòria per segmentació, estarà caracteritzat per:

A) Tenir més memòria ROM que RAM.
B) Tenir la memòria primària més gran que la secundària.
C) La senzillesa a l'hora de compartir zones de memòria primària.
D) Una divisió de la memòria física en segments de longitud fixa.

12.- El 8086 es pot considerar un processador:

A) SISD B) SIMD C) MIMD D) RETI

13.- De les característiques del 8086 podem dir que:

A) Té un sistema d'interrupcions mascarables amb diversos nivells de prioritat.
B) Pot fer accessos a paraula parell o imparell tant a memòria com a E/S.
C) El bus d'adreces és de 32 bits ja que les adreces lògiques són de 32 bits.
D) Cap de les anterios és certa.

14.- En una comunicació série síncrona:

A) La interfície haurà d'estar preparada per transmetre i rebre el senyal de rellotge de la informació.
B) La velocitat de transferència és més gran que en una transmissió per canals d'E/S.
C) La freqüència de mostreig del receptor tolera una desviació gran.
D) Cap de les anteriors és certa.

15.- Si dissenyem una CPU aplicant els conceptes de màquines SIMD:

A) Fariem que cada instrucció només tardés un cicle en executar-se.
B) Les instruccions serien interpretades pel microprograma.
C) Les instruccions seràn executades de forma seqüencial.
D) Tindrem una màquina amb gran capacitat de memòria.

16.- En un sistema basat en el microprocessador 8086, el senyal $\overline{BHE}$ s'utilitza:

A) Com a senyal de gestió de bus.
B) Per adreçar el bloc de bytes en adreça parell quan està a 1.
C) Per indicar que hi ha una adreça vàlida en el bus local.
D) Cap de les anteriors és correcta.

17.- En un programa en ensamblador per a 8086 a partir del qual s'ha d'obtenir un programa executable sobre MS-DOS:

A) Si és de tipus .EXE, l'execució ha d'acabar únicament amb la instrucció:
```
INT 21h.
```
B) La càrrega del programa serà més ràpida que la versió corresponent .COM.
C) L'execució del programa pot acabar amb:
```
MOV AL,0
MOV AH,4CH
INT 21H
```
D) Cap de les anterios és certa.

18.- Es vol implementar un sistema d'E/S on la característica més important és que ocupi el menor temps possible de CPU. Quina afirmació és certa?

A) Utilitzarem la tècnica de consulta d'estat periòdica combinada amb transferència de blocs.
B) Aplicarem interrupcions.
C) La millor solució serà dissenyar un controlador de DMA.
D) Utilitzarem un processador d'E/S per implementar canals d'E/S.

19.- Per poder comunicar un sistema microprocessador amb un terminal remot que està a gran distància, a través d'un enllaç série:

✓ A) Dotarem al sistema microprocessador d'una interfície, basada en el xip de soport 8251A, que, a través d'un modem, enllaça amb el terminal.
B) Dissenyarem un sistema d'E/S basat en el 8255 associant a un dels seus ports un registre de desplaçament.
C) Utilitzarem un canal de DMA per aconseguir la màxima velocitat de transferència.
D) La solució més pràctica seria la que dóna la resposta B, però la transferència hauria de ser, forçosament, síncrona.

20.- En un programa per a 8086 apareix la instrucció CLI:

A) Els registres SI i DI seran incrementats en una instrucció de tractament de string.
B) No seran acceptades les peticions de bus que es facin a partir d'aquest moment.
C) Les interrupcions per condició d'estat no seran ateses.
✓ D) Cap de les anteriors és correcta.

Resposta correcta: 1 punt
Resposta incorrecta: - 1/3 punts
Resposta no contestada: 0 punts

Codi 34013400

durada: 40 minuts

1.- Si es vol dissenyar un processador que es caracteritzi perquè totes les seves instruccions es poden executar en un únic cicle:

A) Farem que la unitat de control sigui microprogramada.
B) Aplicarem conceptes de processadors RISC.
C) Farem que el rellotge del sistema sigui de freqüència elevada.
D) Cap de les anteiors és correcta.

2.- Un sistema amb 8086 vol accedir a la posició física 47702h. Quins valors dels seus registres farien possible aquest accés?

A) DS = 4000h i BX = 7702h
B) ES = 3800h i DI = F702h
C) CS = 4700h i IP = 702h
D) Totes les anteriors són correctes

3.- Quan es vol tenir un sistema de gestió de memòria que permeti obtenir diversos espais d'adreces, ens haurem de basar en:

A) En la segmentació.
B) En la paginació.
C) En la paginació amb tamany de pàgina variable.
D) En al paginació amb tamany de pàgina semivariable.

4.- Si un ordinador disposa d'un processador CISC:

A) La CPU podrà gestionar la memòria per paginació.
B) Els busos d'adreces seran multiplexats amb els de dades.
C) Les instruccions de llenguatge màquina seran variades i potents.
D) La capacitat de les entrades/sortides dependrà del nivell de microprogramació de la unitat de control de la CPU.

5.- Del 8086 podem dir que no és un processador:

A) Matricial. B) Vectorial C) RISC D) Matricial o vectorial o RISC.

6.- En un ordinador amb 8086:

A) Dos processos no es podran executar en paral.lel encara que es compleixi la condició de Bernstein.
B) Si assegurem exclussió mútua entre la regió crítica de dos processos, el recurs compartit es gestionarà correctament.
C) No podem aconseguir multitasca.
D) No podem treballar amb un sistema operatiu multiprocés.

7.- Es vol seguir el funcionament d'un sistema basat en el microprocessador 68000:

A) El millor instrument per aquesta feina és un oscil.loscapi.
B) Es podria utilitzar un analitzador lògic amb una sonda específica per a 8086.
C) Es pot utilitzar un analitzador de xarxes.
D) Es pot utilitzar un analitzador lògic amb una sonda de propòsit general.

8.- Les instruccions més habituals que executarà el TRIN I (ordinador desenvolupat a l'EUPVG) faran la mateixa operació sobre dades diferents. Per fer més ràpid l'ordinador:

A) S'ha fet que tingui un processador vectorial.
B) La CPU només pot ser SISD.
C) No importa el tipus de CPU si es compensa amb una adequada gestió de memòria.
D) Cap de les anteiors és certa.

9.- En la memòria d'un ordinador amb 8086 hi ha les següents dades a partir de la posició 8000h:
47h, 58h, FFh, A0h, 7Fh.
Si la CPU llegeix una paraula en la posició 8001H, podem dir:

A) Es provocarà una interrupció interna per error en l'accés a memòria.
B) S'obtindrà la paraula 5847h.
C) S'obtindrà la paraula 58FFh.
D) S'obtindrà la paraula FF58h.

10.- Si les dades 69h, 70h, 10h, 20h estan en el segment de pila a partir del desplaçament 100h d'un sistema amb 8086, quin valor ha de tenir SP perquè després de la instrucció POP AX, el registre AX = 2010h?

A) 100h B) 102h C) 104h D) 105h

11.- D'un sistema d'E/S no és cert que:

A) Té com a funció adaptar mecànicament el sistema i el perifèric.
B) Potser haurà de tenir capacitat de conversió d'informació.
C) Controlarà la temporització de la comunicació entre CPU i perifèric.
D) La velocitat de procés global del sistema és independent del sistema d'E/S.

12.- Si tenim un sistema d'E/S basat en interrupcions, aconseguirem:

A) Un temps d'atenció mínim.
B) Transferències ràpides i directes entre memòria i dispositiu.
C) Aprofitar més eficientment el temps de CPU.
D) Disminuir l'ocupació dels busos.

13.- La comunicació més ràpida entre un sistema microprocessador de 32 bits i un perifèric de 16 bits que es troba a 20 cm del sistema serà:

A) Una comunicació sèrie combinada amb DMA.
B) Una comunicació sèrie gestionada per consulta d'estat.
C) A través d'un bus de 16 bits.
D) A través d'un bus de 8 bits.

14.- En un programa per a 8086 apareix la instrucció MOV AX,[100h]:

A) El registre de segment que s'utilitza per defecte és el DS.
B) Es fa un accés a la posició 100h del mapa d'entrada/sortida.
C) Es transfereix un byte.
D) Es fa un accés a la posició ES:100h

15.- Al referir-nos a un sistema que està composat per diversos ordinadors però que per l'usuari, es comporta com si fos una sola màquina molt més potent que les màquines que el composen, podriem dir que estem parlant de:

A) Un sistema multiprocessador.
B) Una xarxa d'àrea local.
C) Una màquina MIMD.
D) Un sistema distibuït.

16.- Durant el procés d'enllaçat ("linkat") en un sistema basat en MS-DOS:

A) Es crea un fitxer executable a partir d'un o més fitxers objecte, però els desplaçaments no queden fixats.
B) Es crea un fitxer executable a partir d'un o més fitxers objecte, però si és .EXE les referències a segment no estan fixades.
C) Es crea un fitxer de codi font a partir d'un o més fitxers objecte.
D) Es tradueix el fitxer font a llenguatge màquina.

17.- Sabem que un sistema basat en microprocessador té una unitat d'E/S basat en una interfície de propòsit general. Quina afirmació no és certa?

A) Aquests tipus d'unitat d'E/S facilita la planificació del temps de CPU.
B) Serà fàcil adaptar el sistema a dispositius perifèrics diferents.
C) La programació de la interfície serà complicada.
D) Seria possible disposar d'una comunicació sèrie.

18.- Un sistema microprocessador rep 8 senyals de petició d'interrupció. Quin dels següents mètodes no utilitzarieu per solucionar el problema de les prioritats:

A) Encadenació pipeline.
B) Consulta d'estat.
C) Incloure un controlador d'interrupcions programable.
D) Daisy chain ("cadena margarida").

19.- Es vol dissenyar una unitat d'E/S basada en canals d'E/S. Per aconseguir-ho podriem utilitzar el xip:

A) 8288 B) 8089 C) 8237 D) Cap dels anteriors.

20.- Si un ordinador té un sistema operatiu de processament per lots:

A) La planificació del temps de CPU serà complicada.
B) Serà bo utilitzar-lo en sistemes per a desenvolupament de programes.
C) Hi haurà poca o nul.la interacció entre usuari i programa en execució.
D) Es complica la gestió de memòria.

Resposta correcta: 1 punt **Codi de la prova: 34013500**

Resposta incorrecta: - 1/3 punts

Resposta no contestada: 0 punts durada: 40 minuts

1.- Si es vol dissenyar un processador que es caracteritzi per tenir un conjunt d'instruccions molt variat:

A) Farem que la unitat de control sigui cablejada.
B) Aplicarem conceptes de processadors RISC.
C) Farem que el rellotge del sistema sigui de freqüència elevada.
D) Cap de les anteiors és correcta.

2.- Un sistema amb 8086 vol accedir a la posició física 47702h. Quins valors dels seus registres farien possible aquest accés?

A) DS = 4700h i BX = 0702h.
B) ES = 4800h i DI = F702h.
C) CS = 4070h i IP = 702h.
D) Totes les anteriors són correctes.

3.- Quina de les afirmacions següents no és correcte:

A) Una impressora d'injecció de tinta és lenta, silenciosa i amb gran qualitat d'impressió.
B) Una impressora de línies és tant lenta com una impressora de banda.
C) Una impressora làser es pot considerar una impressora de caràcters.
D) Les impressores matricials poden tenir una velocitat d'impressió de 300cps.

4.- En un sistema basat en un microprocessador 68000 en mode supervisor, quina de les següents adreces no pot correspondre a una paraula (16 bits):

A) 100h B) 10206h C) 000104h D) 0FFDh

5.- El 8086 és un processador de 16 bits que disposa d'una arquitectura caracteritzada, entre altres coses, per:

A) La seva estructura pipeline de dues unitats funcionals.
B) Ser vectorial.
C) Ser RISC amb moltes instruccions de llenguatge màquina.
D) Cap de les anteriors és correcte.

6.- Un ordinador es comunica amb un conjunt de 20 terminals a través d'un canal d'E/S:

A) La CPU estarà dedicada el 99% del temps a feines d'E/S.
B) En el seu disseny s'ha de tenir en compte el problema de competició pel bus.
C) Els programes que s'executen seran més lents.
D) Podem dir que aquest ordinador té una estructura pipeline.

7.- El número de línies que té un monitor de CRT depèn directament dels següents paràmetres:

A) La freqüència de l'oscil.lador vertical.
B) La potència del senyal de vídeo.
C) La separació entre les bobines deflectores.
D) Les freqüències dels oscil.ladors horitzontal i vertical.

8.- La capacitat de memòria física d'un ordinador depèn, únicament,:

A) Del bus d'adreces de la CPU.
B) De l'arquitectura interna de la CPU.
C) D'una adequada gestió de memòria i és independent de la CPU.
D) Cap de les anteriors és certa.

9.- Les característiques següents han estat trobades en una oferta d'ordinadors. Quina és falsa?:

A) Ordinador basat en un 8086 i amb 4MB de RAM.
B) Ordinador basat en un 80386 i amb 1MB de RAM.
C) Ordinador basat en un 68000 i amb 4MB de RAM.
D) Ordinador basat en un 8086 en mode virtual i 8 MB de RAM.

10.- En un ordinador amb un processador SIMD podem dir que:

A) La CPU podrà gestionar la memòria per paginació.
B) Els busos d'adreces seran multiplexats amb els de dades de les ALU.
C) Les instruccions de llenguatge màquina seran variades i potents.
D) Cap de les anteriors és correcte.

11.- Per al disseny d'un controlador per a un torn mecànic s'ha d'escollir el tipus de teclat més adequat, tenint compte que l'entorn de treball a que estarà sotmès es caracteritza per ser humit i brut:

A) Utilitzarem un teclat amb tecles mecàniques.
B) Serà millor utilitzar un teclat de membrana.
C) És més barat utilitzar un teclat amb tecles magnètiques.
D) Cap de les anteriors és certa.

12.- Els tipus de ratolí més usuals són els següents:

A) Mecànics, òptics i magnètics.
B) Mecànics, optomecànics i òptics.
C) Mecànics i capacitius.
D) Optomecànics, magnètics i d'efecte Hall.

13.- Una de les tècniques utilitzades en la codificació de la informació en el procés de gravació sobre disc magnètic és la MFM. Aquesta tècnica consisteix en:

A) Guardar un pols per cada bit amb valor 1, únicament.
B) Inserir un pol entre cada dos polsos de rellotge quan correspon a un bit amb valor 1.
C) Guardar un pols per a cada bit amb valor 1 i un pols de rellotge entre dos bits consecutius amb valor zero .
D) Guardar dos polsos per cada bit amb valor 1 i un pols per cada bit amb valor zero.

14.- El microprocessador 68000 té un conjunt de registres:

A) De 32 bits.
B) Altament especialitzats.
C) Amb accés independent de la part alta o part baixa indistintament.
D) Que són compatibles amb el conjunt de registres del 8086.

15.- Quan es vol tenir un sistema de gestió de memòria que sigui transparent al programador ens haurem de basar:

A) En la segmentació.
B) En la paginació.
C) En la paginació amb tamany de pàgina variable.
D) En al paginació amb tamany de pàgina semivariable.

16.- Al referir-nos a un sistema que està composat per diversos ordinadors però que per l'usuari, es comporta com si fos una sola màquina molt més potent que les màquines que el composen, podríem dir que estem parlant de:

A) Un sistema SIMD.
B) Una xarxa d'àrea local.
C) Una màquina MIMD.
D) Cap de les anteriors és correcte.

17.- El temps d'accés és un dels paràmetres més importants d'una unitat de disc. El seu valor es funció de:

A) Del nombre de sectors de formatat del disc, entre altres factors.
B) Del temps de latència, d'establiment i de recerca.
C) De les ordres que utilitzi la FDC.
D) Del temps de latència i de la velocitat de transferència.

18.- Quina de les següents afirmacions no és correcte:

A) Una unitat de disc magnètic o de CD-ROM fa girar el disc a una velocitat angular constant.
B) La capacitat d'un disc depèn de la densitat de grabació.
C) La velocitat de transferència depèn de la velocitat de rotació del disc.
D) Les unitats de disc magneto-òptic tenen un temps d'accés major que les unitat de disc magnètic.

19.- Si volem obtenir un espai virtual de memòria que sigui pràctic a l'hora d'organitzar separadament els diferents tipus d'informació que conté, aplicarem:

A) Una CPU amb un bus d'adreces de 32 bits.
B) Una gestió de memòria per paginació.
C) Una CPU amb un bus de dades de 32 bits.
D) Una gestió de memòria per segmentació.

20.- Quan s'aplica la tècnica d'entrada/sortida per consulta d'estat en un sistema amb molts perifèrics a controlar, podem assegurar que:

A) El temps de CPU serà ben aprofitat.
B) És imprescindible la utilització d'un controlador de DMA per fer E/S eficients.
C) L'ordre d'atenció a cada perifèric queda definit per l'algorisme de la rutina de servei.
D) Es complica la gestió de memòria.

Resposta correcta: 1 punt **Codi de la prova: 3407401306000**

Resposta incorrecta: - 1/3 punts

Resposta no contestada: 0 punts durada: 40 minuts

1.- Sobre el processament paral.lel, quina afirmació és correcta:

A) Una màquina SIMD es pot considerar un processador paral.lel.
B) El 8086 és un processador paral.lel.
C) En una màquina SISD no es pot aplicar cap grau de paral.lelisme.
D) Una màquina no pot tenir processament paral.lel si no té més d'un processador.

2.- Un sistema amb 8086 vol accedir a la posició física 60703h. Quins valors dels seus registres farien possible aquest accés?

A) DS = 6000h i BX = 0702h.
B) ES = 6070h i DI = 3h.
C) CS = 6003h i IP = 70h.
D) Cap de les anteriors és correcta.

3.- Quina de les afirmacions següents no és correcta:

A) Una impressora d'injecció de tinta és lenta, silenciosa i amb gran qualitat d'impressió.
B) Una tecla de membrana té una vida major que una tecla mecànica.
C) Una impressora làser es pot considerar una impressora de caràcters.
D) Les impressores matricials poden tenir una velocitat d'impressió de 300cps.

4.- Es diu que un microprocessador de 32 bits disposa de gestió de memòria virtual capaç de proporcionar un espai virtual de 64 terabytes. Estan parlant del:

A) 68030 B) 68020 C) 80386 D) Cap dels anteriors

5.- De les afirmacions següents quina no pot ser atribuïda al microprocessador 68000:

A) Té dos modes de funcionament: usuari i supervisor.
B) Té espais d'adreçament separats per a memòria i E/S.
C) La unitat de control és microprogramada i de dos nivells.
D) Disposa del mode d'adreçament indirecte per registre d'adreces amb postincrement.

6.- Un ordinador es comunica amb un conjunt de 20 terminals a través d'un canal d'E/S:

A) La CPU estarà dedicada el 99% del temps a feines d'E/S.
B) En el seu disseny s'ha de tenir en compte el problema de competició pel bus.
C) La CPU ha de consultar cada terminal per saber la seva disponibilitat.
D) Podem dir que aquest ordinador té una estructura pipeline.

7.- Quina de les següents afirmacions no és correcte:

A) Una unitat de disc magnètic o de CD-ROM fa girar el disc a una velocitat lineal constant.
B) La capacitat d'un disc depèn de la densitat de grabació.
C) La velocitat de transferència depèn de la velocitat de rotació del disc.
D) Les unitats de disc magneto-òptic tenen un temps d'accés menor que les unitat de disc magnètic.

8.- La unitat d'interfície amb el bus d'un 8086 serveix per:

A) Adaptar els perifèrics als registres del 8086.
B) Controlar els busos de dades, adreces i control.
C) Fer les transferències entre microprocessador i memòria o E/S.
D) Són correctes la B i la C.

9.- En el 8086 quan es transfereix una paraula amb adreça imparell:

A) El byte menys significatiu passa a través de la part alta del bus de dades.
B) El byte menys significatiu passa a través de la part baixa del bus de dades.
C) No és possible fer una transferència d'aquest tipus.
D) La resposta A és correcta sempre que no hi hagi interrupcions demanades.

10.- Si en un programa per a 8086 volem incloure una instrucció per transferir una dada de memòria a la CPU mitjançant adreçament indirecte per registre, l'operand font es podrà especificar a través del registre:

A) AX B) BX C) CX D) Totes són vàlides

11.- En un sistema amb gestió de memòria per paginació, si s'executa una instrucció que fa una transferència entre CPU i memòria però la pàgina corresponent no està activa:

A) El sistema queda bloquejat.
B) Es fa un reset de la màquina.
C) S'inicia un procés per portar la pàgina sol.licitada des de memòria secundària.
D) El programa corresponent no està ben dissenyat.

12.- Quina de les afirmacions següents no és correcta:

A) La paginació no és transparent al programador.
B) La segmentació permet disposar d'un espai d'adreçament més gran que la memòria principal.
C) La paginació no facilita la compartició de zones de memòria.
D) La segmentació permet la protecció a diferents nivells de les dades de memòria.

13.- En una estructura d'entrada/sortida de tipus radial:

A) El nombre de dispositius controlables està limitat per la quantitat de terminals de la CPU.
B) Es pot emprar per connectar un gran nombre de perifèrics al sistema.
C) Es descarrega a la CPU de les feines d'entrada/sortida.

D) La CPU pot atendre a dos o més perifèrics simultàniament sense fer-los esperar.

14.- El sistema operatiu s'encarrega, entre altres coses, de:

A) Traduir les instruccions de llenguatge màquina a microprograma.
B) Activar els senyals de control dels perifèrics.
C) Facilitar l'ús dels recursos del sistema als usuaris i als programadors, i centralitzar el seu control.
D) Comprovar les línies de petició d'interrupció.

15.- Sobre un monitor de CRT podem dir:

A) L'escombrat de la pantalla es fa combinant desplaçaments horitzontals amb un de vertical.
B) Un monitor color no pot treballar en blanc i negre.
C) Les plaques deflectores del CRT controlen la intensitat de llum del punt il.luminat.
D) Són correctes la A i la C.

16.- Una impressora laser també es pot considerà:

A) Una impressora de caràcteres.
B) Una impressora de línies.
C) Una impressora de pàgines.
D) Cap de les anteriors és correcte.

17.- En un processador matricial és cert que:

A) Hi ha més d'una unitat de control.
B) Executa en paral.lel vàries instruccions.
C) Tracta vàries dades al mateix temps.
D) Cap de les anteriors és correcte.

18.- El temps d'accés és un dels paràmetres més importants d'una unitat de disc. El seu valor és funció :

A) Del nombre de sectors de formatat del disc, entre altres factors.
B) Del temps de latència, establiment i de recerca.
C) Del temps de latècia i de la velocitat de transferència.
D) Cap de les anteriors.

19.- El 68000 es pot considerar un processador:

A) SISD B) RETI C) MIMD D) Cap de les anteriors.

20.- En una sistema basat en el microprocessador 8086, el senyal A0 s'utilitza:

A) Com a senyal de gestió de bus.
B) Per adreçar el bloc de bytes en adreça parell, quan està a 0.
C) Per indicar que hi ha una adreça vàlida.
D) Cap de les anteriors és correcta.

Solucions

PROBLEMA 1

a) El nombre de dades que es transfereixen de memòria a memòria ve donat pel valor del registre CX en el moment de executar-se la instrucció REP MOVSB: **3322H** bytes transferits.

b) SI = 3422H DI = 3472H CX = 0

c) S'ha d'utilitzar el senyal ALE. Quan el senyal ALE doni un pols a nivell 1 indica que en el bus d'adreces hi ha present una adreça.

PROBLEMA 2

CL = 55H

AX = AL x CL = **154H**

DX = 8877H

AL = 0 **AH = 0**

AL = 0 **AH = 0**

PROBLEMA 3

a)
```
MOV AL,40H
OUT 21H,AL
```

b)
```
MOV AX,0
MOV DS,AX
MOV AX,0100H
MOV [100],AX  ; desplaçament
MOV AX,FFOOH
MOV [102],AX  ; desplaçament
```

c) En la posició FFFF0h hi haurà d'haver un salt a la posició FFA00H:

```
...
JMP FAR PTR SALT
...
S SEGMENT AT 0FF00H
        ASSUME  CS:S
        ORG 0A00H
        SALT LABEL FAR
S ENDS
```

d) Escriptura : $M/\overline{IO} = 1$
$\overline{WR} = 0$

Lectura : $M/\overline{IO} = 1$
$\overline{RD} = 0$

e)
```
MOV AX,0
MOV SS,AX
MOV SP,1000H
```

PROBLEMA 4

AX = 050H

AX = 100H

AL = 50H

AL = 03H

AX = 'PI'

AX = 0325H

PROBLEMA 5

a)

```
        MOV SI,OFFSET UNISEG
        MOV DX,0
        MOV CX,6
        TRANS:  MOV AL,[SI]
                OUT DX,AL
                INC SI
                INC DX
                LOOP TRANS
                RET
```

b)

```
        MOV CX,2
        MOV SI,OFFSET UNISEG
SEDIG:  INC [SI]
        CMP [SI],0AH
        JNE  TORNAR
        MOV [SI],0
        INC SI
        INC [SI]
        CMP [SI],6
        JNE TORNAR
        MOV [SI],0
        INC SI
        LOOP SEDIG
        INC [SI]
        CMP [SI+1],2
        JNE NDOS
        CMP [SI],4
        JNE TORNAR
        MOV WORD PTR[SI],0
TORNAR: CALL TRANSFERIEX
        RET
NDOS:   CMP [SI],0A
        JNE  TORNAR
        MOV [SI],0
        INC [SI+1]
        JMP TORNAR
```

c)

```
        PUSH SI
        PUSH DI
        PUSH AX
        PUSH CX
        PUSH DX
        DEC  COMPTADOR
        JNZ  RETORN
        MOV  COMPTADOR,1000
        CALL  ACTUALITZAR
RETORN: POP DX
        POP CX
        POP AX
        POP DI
        POP SI
        IRET
```

d) La transferència hauria de ser mode burst, ja que seria el tipus d'accés que demanaria menys vegades els busos.

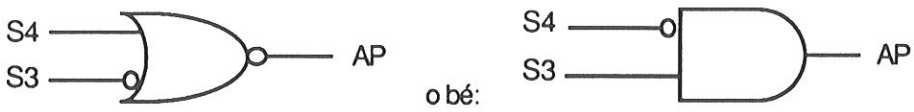
PROBLEMA 6

a) A i B han d'indicar un accés a pila => A,B = S3,S4 amb:

A = S3 = 1

B = S4 = 0

Llavors:

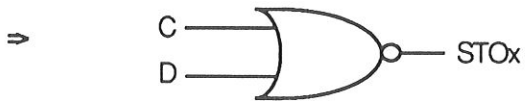


Se suposa que el detector tindrà informació per saber quan S3 i S4 són presents en els terminals del 8086.

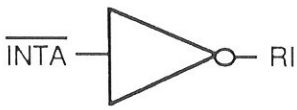
b) Per poder accedir al detector en operacions de lectura :

lectura : $C = M / \overline{IO}$

$D = \overline{WR}$ -> STOx = 1



c) E ha d'estar relacionada amb $\overline{INTA}$ donat que aquest és actiu a nivell baix i RI ho és a nivell alt:



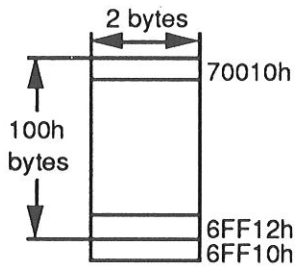
d) Adreça de la Pila: 70010h => part alta de la pila (primers elements introduïts)

longitud: 100h bytes.

L'adreça de l'últim element introduït:

70010h - 100h + 2 = 6FF12h

Com que el detector genera una interrupció quan el valor del bus d'adreces és menor que l'adreça referència que té guardada, en els registres REG1 i REG2 s'haurà de guardar l'adreça de l'últim element introduit:



6FF12h -> seg = 6FF1h
desp = 0002h
(Per exemple)

Finalment, només s'ha de tenir en compte que SS i SP s'han d'inicialitzar amb valors tals que SP es pugui decrementar 100h vegades sense passar de zero. També es considera que la primera operació que es fa a la pila és una introducció i llavors:

SS = 6FF1h

SP = 0102h , això farà que el primer element s'introdueixi en la posició 70010h.

La seqüència d'instruccions:

```
MOV AX,6FF1h        ; canvia la pila
MOV SS,AX
MOV AX,102H
MOV SP,AX
MOV AX, 6FF1h       ; carrega límit al detector
OUT 10H,AX          ; seg.
MOV AX,102h
OUT 14h,AX          ; desp
```

e) La seqüència d'inicialització haurà de donar valors de vector d'interrupció a la taula de vectors d'interrupció i també haurà de carregar el codi de tipus en el detector.

Adreça subrutina d'atenció: 8000:0100h

Codi de tipus: 82h

L'adreça del vector pel codi de tipus 82h:

82 x 4 = 208 => desp: 208h , seg: 20Ah

La seqüència d'instruccions podria ser la següent:

```
MOV AX,0          ; inicialitza vector en la taula
MOV DS,AX
MOV AX,100h
MOV [208H],AX
MOV AX,8000H
MOV [20AH],AX
MOV AL,82H        ; inicialització codi de tipus
OUT 12h,AL        ; en el detector
```

PROBLEMA 7

a) CS:103h = FFh

CS:104h = 00h

CS:105h = 'P'

...

CS:204h = 'P'

b) END ENTRADA no s'executa mai doncs és una directiva d'ensamblador.

L'etiqueta ENTRADA serveix per indicar a l'ensamblador i al linkador el punt d'entrada al programa, o sigui, la primera instrucció que s'executarà.

c)

BL = 80h

AL = 04h

DI = 00FFh

d) Quan es cridessin els procediments es guardaria CS i IP a la pila i quan es retornés es recuperarien dues paraules de la pila => l'execució seria més lenta.

PROBLEMA 8

a)
CX = 8
SI = 36H
DI = 100H
DS = 4000H
ES = 5000H
BX = 30H

b)
SI = 39H
DI = 102H
DS = 4000H
ES = 5000H

c)
SI = 36H
DI = 104H
CX = 8H

d)
SI = 3EH
DI = 108H
CX = 0

50100H =	02H
	56H
	18H
	26H
	03H
	89H
	13H
50107H =	23H

PROBLEMA 9

a)
MOV SS,[104H]
MOV SP,[106H]

b)
DS:104H = 1000H
106H = 00FAH
108H = 2000H
10AH = 00FEH
10CH = 5FFEH
10EH = 6000H

c)
100FFH = 10H
100FEH = 50H
100FDH = 70H
100FCH = 00H
100FBH = 01H
100FAH = 50H

d)
200FFH = 02H
200FEH = 52H

e)
65FFFH = AH
65FFEH = AL

PROBLEMA 10

```
CONVERSIO PROC
          PUSH BX
          PUSH AX
          STD
          ADD DI,CX
          DEC DI
          MOV BX, OFFSET TAULA_CARACTERS
CONV:     MOV AL, [SI]
          AND AL, 0FH
          XLAT
          STOSB
          MOV AL, [SI]
          AND AL, 0F0H
          SHL AL, 1
          SHL AL, 1
          SHL AL, 1
          SHL AL, 1
          XLAT
          STOSB
          INC SI
          LOOP CONV
          POP AX
          POP BX
          RET
CONVERSIO ENDP
```

PROBLEMA 11

A) Mode màxim.

B) Per a memòria : 20 bits d'adreça => 1Mbyte

Per a E/S : 16 bits d'adreça => 64Kbytes

C) Tots els registres de segment podrien tenir el mateix valor i per tant tindriem tots els segment sobre la mateixa zona de memòria.

D) A l'hora de transferir paraules entre memòria i CPU 8086 les paraules que tenen adreça parell són transferides amb un sol cicle de bus, mentre que les paraules amb adreça imparell necessiten dos cicles de bus per ser transferides.

PROBLEMA 12

Es pot pensar que hi haurà moltes consultes que no trobaran cap byte disponible en el perifèric i per tant serà temps perdut. Si volguéssim solucionar això fent que les consultes fossin més separades en el temps hi hauria el perill de saturar el buffer del perifèric si la informació no fos llegida a temps.

Per tant, donades les possibilitats de produir interrupcions que té el perifèric, la solució més convenient al problema que planteja el prototipus seria comença la comunicació amb una interrupció que hauria de produir el perifèric quan tingués al menys un bytes disponible, fent que el 8086 passés a executar la rutina d'atenció. D'aquesta menera el 8086 evita les consultes inecessàries no desaprofita temps d'execució.

Problema 13

Part A

A1) MOV DX, **301H** ; (2)
MOV DX, **300H** ; (7)

91T T=0'2μs

A2) $t_{ma} = t_{11} + t_{12} + t_{13} + t_{14} + t_1 + t_2 + t_3 + t_4 + t_{5f} + t_6 + t_7 + t_8 = 91$ cicles = 18,2 μs

A3) El periode de transferència i la velocitat de transferència són:

$$t_r = \sum_{i=1}^{10} t_i + \sum_{i=13}^{16} t_i = 63 + 55 = 118 \text{ cicles} = 23{,}6\ \mu s$$

$$v = 1/t_r = 42372{,}8 \text{ bytes/s}$$

Part B

B1) (A) MOV AL, **60H**
(B) MOV SI, **180H** ; 60H * 4
(C) MOV [SI], OFFSET **RUSIMP**
(E) MOV DX, **301H**
(22) MOV DX, **300H**

B2) STI => IF <— 1
Permet que el 8086 atengui interrupcions mascarables.

B3) El periode de transferència és:

$$t_r = t_{rut} + t_{resp}\Big|_{t_i=0} = \sum_{j=17}^{29} t_j + 68 = 140 + 68 =$$

68

$$= 208 \text{ cicles} = 41{,}6\ \mu s$$

La velocitat de transferència:

$$v = 1 / t_r = 1 / 41{,}6\mu s = 24038{,}4 \text{ bytes/s}$$

B4) Una millorar del disseny en aquestes condicions podria consistir en fer consultes d'estat dins de la rutina d'atenció a la interrupció que genera la interfície, de manera que quan la interfície deixés d'estar disponible durant la seva atenció es suspengués la transferència i es retornés de la rutina de servei a la interrupció.

Per aconseguir això la rutina d'atenció es podria modificar de la següent manera:

```
RUSIMP  PROC    FAR
        PUSH    DX
        PUSH    BX
        PUSH    AX
        MOV     BX, PUNT
CONTTR: MOV     AL, DATIMP[BX]
        MOV     DX, 300H
        OUT     DX, AL
        INC     BX
        MOV     DX, 301H  ; comprovar si interfície disponible
        IN      AL, DX
        TEST    AL, 01H
        JNZ     CONTTR          ; si disponible continuar la transferència
        MOV     PUNT, BX
        POP     AX
        POP     BX
        POP     DX
        IRET
RUSIMP  ENDP
```

PROBLEMA 14

PART A

A1) La instrucció 2: IN AL, 31H

La instruccció 6: OUT 30H, AL

La instrucció 7: POP AX

A2) f = 5MHz => T = 0,2μs

$t_{ma} = t_9 + t_{10} + t_{11} + t_{12} + t_1 + t_2 + t_3 + t_{4f} + t_5 + t_6 = 4 + 4 + 17 + 28 + 11 + 10 + 4 + 4 + 2 + 10 = 94$ cicles ⇒ **18,8μs**

$t_r = t_7 + t_8 + t_{11} + t_{12} + t_{13} + t_{14} + t_{15c} + t_1 + t_2 + t_3 + t_{4f} + t_5 + t_6 =$
$= 8 + 18 + 17 + 28 + 2 + 3 + 16 + 11 + 10 + 4 + 4 + 2 + 10 = 133$ cicles $\Rightarrow$ **26,6μs**

A3) $t_{línia} = t_{ma} + (N-1)t_r = 18,8 + (N-1)26,6 =$ **(26,6N - 7,8)μs**

PART B

B1)

a) $t_{ma1} = t_{16} + t_{17} + t_{18} + t_{19} + t_{20f} + t_{21} + t_{22} + t_{23f} + t_{24} + t_{25} + t_{26} = 4 + 4 + 2 + 4 + 4 +$
$+ 10 + 4 + 4 + 17 + 2 + 10 = 65$ cicles $\Rightarrow$ **13,0μs**

Per a la interfície 2, suposant que el primer caràcter que transfereix la interfície 1 no és un CR i no està disponible en la següent transferència:

$t_{ma2} = t_{ma1} + (t_{27} + t_{28c} + t_{30}) + (t_{19} + t_{20f} + t_{21} + t_{22} + t_{23c}) + (t_{31} + t_{32f} + t_{33} + t_{34} +$
$+ t_{35f} + t_{36} + t_{37} + t_{38}) = 65 + (3 + 16 + 15) + (4 + 4 + 10 + 4 + 16) + (4 + 4 + 10 + 4 + 4 +$
$+ 17 + 2 + 10) = 65 + 34 + 38 + 55 = 192$ cicles $\Rightarrow$ **38,4μs**

Per a la interfície 2, suposant que el primer caràcter que transfereix la interfície 1 és un CR:

$t_{ma2} = t_{ma1} + (t_{27} + t_{28f} + t_{29} + t_{30}) + (t_{19} + t_{20c}) + (t_{31} + t_{32f} + t_{33} + t_{34} + t_{35f} +$
$+ t_{36} + t_{37} + t_{38}) = 65 + (3 + 4 + 4 + 15) + (4 + 16) + 55 = 65 + 26 + 20 + 55 =$

$= 166$ cicles $\Rightarrow$ **33,2μs**

Per a la interfície 2, suposant que el primer caràcter que transfereix la interfície 1 no és un CR i està disponible en les següents transferències fins al final de la transferència de la línia:

$t_{ma2} = t_{ma1} + (N - 1)(t_{27} + t_{28c} + t_{30} + t_{19} + t_{20f} + t_{21} + t_{22} + t_{23f} + t_{24} + t_{25} + t_{26}) +$
$+ (t_{27} + t_{28f} + t_{29} + t_{30}) + (t_{19} + t_{20c}) + (t_{31} + t_{32f} + t_{33} + t_{34} + t_{35f} + t_{36} + t_{37} + t_{38}) = 65 +$
$+ (N - 1)(3 + 16 + 15 + 4 + 4 + 10 + 4 + 4 + 17 + 2 + 10) + 26 + 20 + 55 = 166 + 89(N - 1) =$
$= (33,2 + 17,8(N - 1))$μs $=$ **(17,8N + 15,4)μs**

b) $t_{ma1b} = t_{ma1} = 65$ cicles $\Rightarrow$ **13,0μs**

Per a la interfície 2, suposant que el primer caràcter que transfereix la interfície 1 no és un CR i que aquest dispositiu no està disponible en la següent transferència:

$$t_{ma2} = t_{ma1b} + (t_{27} + t_{28c} + t_{30}) + (t_{31} + t_{32f} + t_{33} + t_{34} + t_{35f} + t_{36} + t_{37} + t_{38}) =$$
$$= 65 + (3 + 16 + 3) + (4 + 4 + 10 + 4 + 4 + 17 + 2 + 10) = 142 \text{ cicles} \Rightarrow \mathbf{28{,}4\mu s}$$

Per a la interfície 2, suposant que el primer caràcter que transfereix la interfície 1 és un CR:

$$t_{ma2} = t_{ma1b} + (t_{27} + t_{28f} + t_{29} + t_{30}) + (t_{31} + t_{32f} + t_{33} + t_{34} + t_{35f} + t_{36} + t_{37} + t_{38}) =$$
$$= 65 + (3 + 4 + 4 + 3) + (4 + 4 + 10 + 4 + 4 + 17 + 2 + 10) =$$
$$= 65 + 14 + 55 = 134 \text{ cicles} \Rightarrow \mathbf{26{,}8\mu s}$$

Per a la interfície 2, suposant que el primer caràcter que transfereix la interfície 1 no és un CR i està disponible en les següents transferències fins al final de la transferència de la línia:

$$t_{ma2} = t_{ma1b} + (t_{27} + t_{28c} + t_{30}) + (t_{31} + t_{32f} + t_{33} + t_{34} + t_{35f} + t_{36} + t_{37} + t_{38}) =$$
$$= 65 + (3 + 16 + 3) + (4 + 4 + 10 + 4 + 4 + 17 + 2 + 10) = 142 \text{ cicles} \Rightarrow \mathbf{28{,}4\mu s}$$

Part C

C1)

(A) MOV AL, 42H

(B) MOV SI, 42H*4

(C) MOV [SI], OFFSET RUT_I

(D) MOV [SI + 2], SEG RUT_I

(E) OUT 32H, AL

C2)

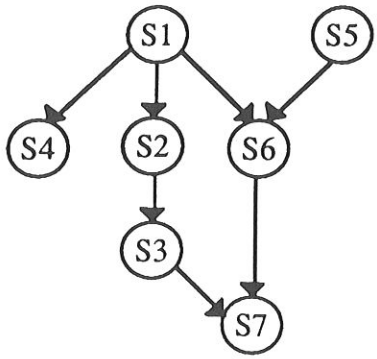
$$t_{ma} = t_{resp} + t_{43} + t_{44} + t_{45} + t_{46} + t_{47} = (68 \text{ a } 252) + (11 + 11 + 14 + 17 + 10) =$$
$$= (68 \text{ a } 252) + 63 = 131 \text{ a } 315 \text{ cicles} \Rightarrow \mathbf{(26{,}2 \text{ a } 63)\mu s}$$

$$t_r = t_{resp} + t_{rut} = 68 + (t_{43} + t_{44} + t_{45} + t_{46} + t_{47} + t_{48} + t_{49} + t_{50} + t_{51} + t_{52}) =$$
$$= 68 + (11 + 11 + 14 + 17 + 10 + 2 + 15 + 8 + 8 + 24) = 68 + 120 = 188 \text{ cicles} \Rightarrow \mathbf{37{,}6\mu s}$$

$$v = 1/t_r = 1/37{,}7 = \mathbf{26595{,}7 \text{ bytes/s}}$$

Problema 15

Graf de precedència i assignació de sentències a processadors:



	P1	P2	P3	P4
F1	S1	S2	S3	S4
F2	S5	S2	S3	S4
F3	S6	S7	S3	
F4		S7		

Fi : fase de processament i; cada fase té una durada de 20 unitats de temps. En la taula es representa l'assignació de sentències a cada processador i el temps que passa fins que finalment són executades. Per tant, tal com es veu en la taula el processador que més temps té la mateixa instrucció és el 3. Això vol dir que serà el que inverteix més temps en esperes de precedència.

Problema 16

f = 5MHz => T = 200ns.

Cas més favorable: Començant l'execució per la instrucció IN fins que es fa el salt cap a la rutina d'atenció a la impressora considerant que els altres dispositius no necessiten atenció.

$t_{esp} = (10 + (2 + 4) * 7 + 2 + 16) * T = 70T$ => $t_{esp} = 14\ \mu s$

Cas més desfavorable: Seria el cas en que tots els dispositius demanessin atenció.

$t_{esp} = (10 + (2 + 16 + 100) * 7 + 2 + 16) * T = 854T$ => $t_{esp} = 170{,}8\ \mu s$

Es podria considerar un altre cas desfavorable, on es consideraria que tots els dispositius estan disponibles, excepte la impressora, que passarà a aquest estat just després de ser executada la instrucció IN per primera vegada. Aquí també suposem que tots els altres dispositius tornaran a estar disponibles a la segona consulta.

$t_{esp} = (10 + (2 + 16 + 100) * 7 + 2 + 4 + 15 + 10 + (2 + 16 + 100) * 7 + 2 + 16) * T = 1711T$

=> $t_{esp} = 342{,}2\ \mu s$

Problema 17

Es tracta d'un problema similar al del conegut com a problema del productor consumidor:

```
N : longitud de la cua
var acua1, cua1_buida, cua1_plena : semàfor;
var p: paquet;
var destí : adreça;
acua1 := inisem(1);
cua1_plena := inisem(0);
cua1_buida := inisem(N);
```

```
Procedure receptor;
...
obtenir_destinació(p,destí);
if destí = camí1 then begin
wait(cua1_buida);
wait(acua1);
   posar_paquet_cua1;
signal(acua1);
signal(cua1_plena);
end;
...
```

```
procedure emissor;
...
wait(cua1_plena);
wait(acua1);
   extreu_paquet_cua1;
signal(acua1);
signal(cua1_buida);
   enviar_paquet_pel_camí1
...
end;
```

Problema 18

-A-

La taula de vectors d'interrupció, en el 8086, està situada a partir de la posició física 0 i ha de contenir les adreces de les rutines d'atenció associades a cada tipus d'interrupció:

```
...
MOV AX,0
MOV DS, AX
MOV DI, 50H*4
MOV [DI], OFFSET RUT_CINTA
MOV [DI + 2], SEG RUT_CINTA
MOV [DI + 4], OFFSET RUT_ADQ
MOV [DI + 6], SEG RUT_ADQ
STI
...
```

-B-

Es tracta d'inicialitzar el registre de control del controlador d'interrupcions perquè accepti peticions d'interrupció des de qualsevol de les seves dues entrades i també s'haurà d'inicialitzar el registre de vector d'interrupció amb el codi de tipus que haurà de donar quan rebi un INTA.

```
...
MOV AL, 3
OUT 62h, AL
MOV AL,50H
OUT 64H,AL
...
```

-C-

El buffer de 2KB s'omplirà deprés de fer 128 lectures del sistema d'adquisició, ja que en cada lectura s'obtenen 16 bytes. Per tant, podrem saber que el buffer és ple quan la variable N_LECT arribi al valor de 128. Per guardar la informació del sistema d'adquisició s'ha d'utilitzar el valor de la variable P_DADES_ADQ que contindrà el desplaçament de la següent posició lliure dins del buffer, on es podrà carregar els valors donats pels conversors.

La preparació del sistema de cinta consistirà en programar el registre de control del DMAC per poder fer les transferències per DMA en mode Burst, de sortida, amb petició d'interrupció, amb increment automàtic del registre d'adreces del DMAC a cada transferència DMA, inicialització del registre d'adreces i del registre comptador, etc.

```
            N_LECT     DB          0
            DADES_ADQ              DB      2048 DUP(0)      ; buffer
            P_DADES_ADQ DW         OFFSET  DADES_ADQ        ; punter al buffer

RUT_ADQ  PROC
            PUSH CX
            PUSH DI
            PUSH AX
            PUSH DX
            PUSH BX
            STI
            MOV CX, 8
            MOV DI, P_DADES_ADQ
            MOV DX, 10H
LECTURA:    IN AL,DX                          ; Lectura dels ADC
            MOV AH, AL
            IN AL, DX
            XCHG AL, AH                       ; ordenar part alta i baixa
            MOV [DI], AX
            ADD DI, 2                         ; actualitzar punters
            ADD DX, 2
            LOOP LECTURA
            MOV P_DADES_ADQ, DI               ; guardar punter
```

```
                INC N_LECT                          ; actualitzar la variable N_LECT
                CMP N_LECT, 128                     ; màxim de lectures possibles
                        ; comprovar si és la última lectura (=> buffer ple)
                        ; si és ple saltar a preparar transferència a cinta
                        ; sino retornar
                JE PRE_CINTA
                POP BX
                POP DX
                POP AX
                POP DI
                POP CX
                IRET
PREP_CINTA:                                 ; preparar transferència a cinta
                MOV P_DADES_ADQ, OFFSET DADES_ADQ       ; Reset del punter
MIR_EST:        IN AL, 32H                  ; Cinta està disponible?
                TEST AL, 1
                JNZ MIR_EST                 ; si lliure, preparar
                MOV AL, 1EH                 ;programar mode burst,incre. adreces,pet. int., reco. DMA,...
                OUT 40H, AL
                MOV CL,4  ; calcular adreça física de memòria per fer trans. per DMA
                MOV DX, SEG DADES_ADQ
                MOV BL, 0
MUL16:          SHL DX, 1
                RCL BL,1
                LOOP MUL16
                ADD DX, OFFSET DADES_ADQ
                ADC BL, 0
                MOV AL, DL                  ; inicialitzar registre d'adreces DMA, etc.
                OUT 46H, AL
                MOV AL, DH
                OUT 46H, AL
                MOV AL, BL
                OUT 50H, AL
                MOV AL, 0                   ; inicialitzar registre comptador DMA
                OUT 44H, AL
                MOV AL, 1
                OUT 44H, AL
                MOV AL, 1                   ; inicialitzar registre control interfície i ordenar transf.
                OUT 30H, AL
                POP BX
                POP DX
                POP AX
                POP DI
                POP CX
                IRET
RUT_ADQ ENDP
```

- D -

En aquest apartat només calia comprovar si s'havia produït algun error en la transferència i reinicialitzar el registre comptador del DMAC. També s'ha de comprovar si s'ha acabat la transferència de tot el buffer.

```
RUT_CINTA PROC
        PUSH AX
        IN AL, 32H          ; comprovar errors
        TEST AL, 2
        JNZ ERROR_PAR
        TEST AL, 4
        JNZ ERROR_CINTA
```

```
                INC N_BLOC          ; actualitzar el comptador de blocs transferits
                CMP N_BLOCS, 8      ; s'han fet totes les transferències de bloc
                JE TRANS_ACABADA
                MOV AL, 0                   ; preparar nova transferència
                OUT 44H, AL
                MOV AL, 1
                OUT 44H, AL
                MOV AL, 1
                OUT 30H, AL
                POP AX
                IRET
TRANS_ACABADA:
                MOV AL, 0                   ; "desactivar" DMA
                OUT 40H, AL
                POP AX
                IRET
RUT_CINTA ENDP
```

-E-

El temps de transferència de tot el buffer (t_{tran}) serà:

t_{tran} = 8*(temps per a cada transferència de bloc) + 7*(temps per cada processament de la rutina RUT_CINTA)

Temps per a cada transferència de bloc:

Velocitat del dispositiu: 100KB/s => una transferència de byte cada 10 µs.

Petició i retorn dels busos: 2 * 2T = 4T = 0,8 µs

Transferència DMA del bloc: 256 * (10 µs) = 2560 µs

Total: **2560,8 µs**

Temps per a cada processament de la rutina RUT_CINTA:

$t_{resp} + t_{rut}$ = (68 a 252)T + 112T = (180 a 364)T = (36 a 72,8) µs

Llavors:

t_{tran} = 8*(2560,8) + 7*(36 a 72,8) ≈ 8*(2560,8) = 20486,4 µs = **20,5 ms**

-F-

Per mostrejar a 1kHz hauriem de ser capaços de fer una lectura cada 1 ms. Però, hem vist en l'apartat anterior que es necessiten 20,5 ms per transferir les dades obtingudes en l'anteior mostreig i per tant, amb la configuració proposada del sistema no es podrà mostrejar a 1KHz.

Per solucionar-ho, hem de tenir en compte que el temps de transferència que hem trobat anteriorment és degut, funamentalment, a la velocitat de transferència de la cinta. Haurem de dotar al sistema de dos buffers de dades per tal que mentre en un es guarda la informació d'una nova adquisició, l'altre serà transferit a la cinta. Així, ara es disposarà de 8 ms per poder transferir el buffer, i, per tant, haurem de disposar d'una unitat de cinta més ràpida. També caldrà realitzar el DMA en mode byte per tal que la CPU pugui utilitzar els busos mentre en fa la transferència sobre la cinta, d'altre manera tampoc es podria complir l'especificació de mostreig.

-G-

Per calcular el paràmetre M cal estudiar la rutina NUCLI:

$t_{mos} = (((17*(2^{16} - 2) + 5) + 8 + 4 + 8) * M + 17 * (M-1) + 5 + 4 + 4 + 4 + 8 + 16) * T = 1\ s$

$t_{mos} = ((17 * 2^{16} + 8) * M + 24) * T = 1\ s$

$$M = \frac{(t_{mos} * f - 24)}{(17*2^{16} + 8)} = \frac{1 * 5 \cdot 10^6 - 24}{(17*2^{16} + 8)} = 4{,}48$$

=> M = 4

(també es podria considerar M = 5, en funció de altres condicionants del sistema).

Problema 19

```
var control: semàfor;
control := inisem(20);
...
/* regió crítica d'accés a la base de dades */
wait(control);
   accés_base();
signal(control);
/* final de la regió crítica */
```

Problema 20

A)

```
...
MOV AX,0
MOV DS, AX
MOV DI, 50H*4
MOV [DI], 500H
MOV [DI + 2], 7000H
MOV [DI + 4], 0
MOV [DI + 6], 800H
MOV [DI + 8], 4H
MOV [DI + 10], 1000H
...
```

B) Aquest conjunt d'instruccions hauria de formar part dels programes d'inicialització del sistema o, com a mínim, dels programes que preparessin al sistema per gestionar als tres perifèrics, i sempre abans que es permetés la petició d'interrupcions d'aquest dispositius, ja que la informació de la taula de vectors ha de servir perquè el 8086 pugui trobar, automàticament, la rutina d'atenció a partir del codi de tipus proporcionat pel dispositiu.

Problema 21

A) Cal trobar la velocitat de transferència que es pot aconseguir:

$$f = 5MHz \Rightarrow T = 200ns.$$

El període de transferència:

$$T_t = (10 + 3 + 4 + 10 + 10 + 16)T = 53T = 10{,}6\ \mu s$$

i la velocitat de transferència màxima que es pot aconseguir:

$$v_t = 1 / (10{,}6\ \mu s) = 0{,}0943 \times 10^6 = 94\ Kbyte/s$$

94 Kbyte/s < 250 Kbyte/s => No es pot emprar la tècnica de consulta d'estat ja que no podem atendre al dispositiu a nivell de velocitat de transferència.

B)

```
        ...
        MOV CX, 4          ; Càlcul de l'adreça física.
        MOV DX, 7000H
        MOV BL, 0
MUL16:  SHL DX, 1
        RCL BL,1
        LOOP MUL16
```

```
ADD DX, 100H
ADC BL, 0
MOV AL, DL          ; inicialitzar registre d'adreces DMA, etc.
OUT 46H, AL
MOV AL, DH
OUT 46H, AL
MOV AL, BL
OUT 50H, AL
MOV AL, 0           ; inicialitzar registre comptador DMA
OUT 44H, AL
MOV AL, 1
OUT 44H, AL
MOV AL, 011111B             ; inicialitzar registre de control de DMAC
OUT 40H,AL
...
```

C) Amb la tècnica anterior es podria aconseguir com a màxim la velocitat de transferència següent:

T_t = Transferència 1 byte = 3T = 0,6 µs

v_t = 1/ (0,6 µs) = 1,66 Mbyte/s

Problema 22

A)

```
RUT_SORT PROC
            PUSH AX
CONSUL:  IN AL, 34H
            TEST AL, 1
            JZ CONSUL
            MOV AL, DADA
            OUT 32H, AL
            POP AX
            RET
RUT_SORT ENDP
```

B)

f = 5 MHz => T = 0,2 µs

t_{ma} = (8 + 8 + 4 + 4 + 8 + 8)T = 40T = 8 µs

C) Considerem que el temps de crida i d'altres processos que pugui fer el programa principal són menyspreables (no sempre serà bona aquesta aproximació, però aquí la fem per simplificar):

t_t = (8 + 8 + 4 + 4 + 8 + 8 + 8 + 18)T = 66T = 13,2 µs

v = 1 / 13,2 µs = 75,757 KB/s (velocitat màxima teòrica)

$t_{rd} = 400\ \mu s$

$t_d = (8 + 18 + 8)T = 34T = 6{,}8\ \mu s,$

això vol dir que: $t_{rd} >> t_d$, i llavors:

$v_r = 1 / 400 = 2{,}5$ KB/s (velocitat real de transferència)

Problema 23

A)

```
...
PUSH DS
MOV AX,0
MOV DS, AX
MOV BX, 68h*4
MOV [BX], OFFSET RUT_SER
MOV [BX + 2], SEG RUT_SER
POP DS
MOV AL, 68h
OUT 58h, AL
MOV AH, 1
OUT 56h, AL
STI
...
```

B) Considerem que la variable PUNT_D conté un desplaçament relatiu a la variable DADES_I. Dins de la rutina de servei a la interrupció es fa una transferència per consuta d'estat per aprofitar els avantatges que ens aporta el buffer de la impressora (major velocitat de transferència). Aquesta opció és vàlida ja que el temps de resposta, t_{rd}, del dispositiu és petit respecte al temps disponible, t_d, que ens dóna la rutina, i respecte al període de transferència per al cas que la rutina només transferís un byte cada vegada.

```
RUT_SER PROC
        PUSH BX
        PUSH AX
        PUSH CX
        MOV BX, PUNT_D
        MOV AL, [BX]
        INC BX
        OUT 52h, AL
        MOV CX, 99
CONEST: IN AL, 54h
        TEST AL, 1
        JZ CONEST
        MOV AL, [BX]
        OUT 52h, AL              ; (1)
        INC BX
        LOOP CONEST
        MOV PUNT_D, BX
        POP CX
```

```
                POP AX
                POP BX
                IRET
RUT_SER
```

C) t_t és el temps entre dues execucions consecutives de la instrucció (1):

$t_t = 4 + 16 + 8 + 4 + 4 + 8 + 8 = 52T = 10{,}4\ \mu s$

$v = 1 / 10{,}4\ \mu s = 96{,}153$ KB/s

Problema 24

A)

```
MOV AL, 011101B        ; inicialitza el registre de control del DMAC
OUT 60H, AL
MOV AL, 0              ; inicialitza el registre comptador del
DMAC
OUT 64H, AL
MOV AL, 2
OUT 64H, AL
MOV AL, 0              ; inicialtiza el registre d'adreces del
DMAC
OUT 66H, AL            ; amb l'adreça física 44000h
MOV AL, 40h
OUT 66H, AL
MOV AL, 4              ; inicialitza latch que conté els 4 bits alts
de l'adreça
OUT 70H, AL
STI
```

B)

t_t = Transferència 1 byte = 4T = 0,8 µs

$v_t = 1/ (0{,}8\ \mu s) = 1{,}25$ Mbyte/s

Problema 25

A) Si un perifèric demana una interrupció durant un procés de DMA s'haurà d'esperar. Hem de comprovar que aquest temps d'espera no sigui superior al temps màxim d'espera de reconeixement a una petició: 100 µs.

$t_{espera} = 3\ \mu s + 512*(1 / 1\ MB/s) + 3\ \mu s + tresp + tma =$

$= 3 + 512 + 3 + 68*0{,}2 + 20 = 551{,}6\ \mu s \Rightarrow 551{,}6\ \mu s > 100\ \mu s$

=> el perifèric no serà atès adequadament.

B) Partint d'un $t_{espera} = 100\ \mu s$, volem trobar la longitud màxima de bloc, N:

$t_{espera} = 3\ \mu s + N*(1 / 1\ MB/s) + 3\ \mu s + tresp + tma = 100\ \mu s$

$3 + N + 3 + 68*0{,}2 + 20 = 100\ \mu s$

$100 = 39{,}6 + N$

$N = 60{,}4 \qquad \Rightarrow N = 60$ bytes

Problema 26

A)

El port de dades té assignada l'adreça 22h

El port d'estat té assignada l'adreça 20h

La consulta d'estat es fa per bloqueig de programa ja que la consulta es manté fins que el dispositiu es troba disponible.

B)

Per cada crida de la rutina es transfereix un byte.

C) Considerem que el temps de crida i d'altres processos que pugui fer el programa principal s'inclouent en T_S:

$f = 5$ MHz => $T = 0{,}2\ \mu s$

$t_t = (8 + 18)T + T_S + (8 + 8 + 4 + 4 + 8 + 8)T = 66T + T_S$

Suposant $T_S = 0$:

$t_t = 66T = 13{,}2\ \mu s$

$v = 1 / 13{,}2\ \mu s = 75750$ B/s

Suposant $T_S = (15 + 35)T = 50T$, on 35T corresponen a feines que ha de fer el programa principal per preparar la dada següent i colocar-la en la variable VALOR:

$t_t = 66T + 50T = 116T = 23{,}2\ \mu s$

$v = 1 / 23{,}2\ \mu s = 43103{,}4$ B/s

D) La impressora accepta un byte cada 1/200 s = 0,005 s = 5 ms. Com que 5 ms és molt més gran que 0,0132 ms o que 0,0232 ms podem aproximar:

$t_t = 5$ ms

$v = 1/5$ ms = 200 B/s

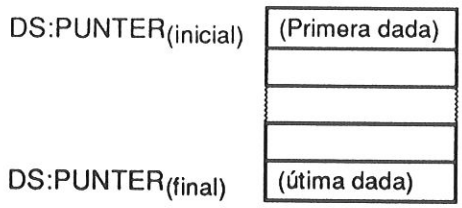
Problema 27

A) El registre de control de la interfície haurà de tenir el valor 11000001 = C1h, que s'utilitza com a codi de tipus d'interrupció, llavors:

```
...
PUSH AX
PUSH DS
```

```
MOV AX,0
MOV DS, AX
MOV BX, 0C1h * 4
MOV [BX], 100H
MOV [BX + 2], 4020H
MOV AL, 1h
OUT 44h, AL
STI
POP DS
POP AX
...
```

B) La variable PUNTER guarda l'adreça de la posició de memòria disponible per guardar una nova entrada, un nou byte procedent del teclat. Les dades es guarden en la zona:



C)

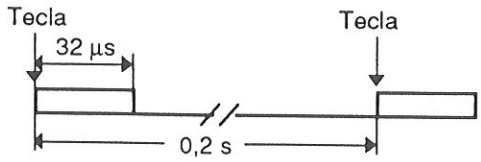
$f = 5$ MHz => $T = 0{,}2\ \mu s$

$t_t = 8 + 4 + 8 + 8 + 8 + 24 + 68 + 8 + 8 + 8 + 8 = 160T = 32\ \mu s$

$v = 1 / 32\ \mu s = 31250$ B/s

D)

Una tecla cada 60/300 = 0,2 s, i una transferència dura 32 µs :



Llavors, el tant per cent que la CPU està dedicada a les entrades per teclat:

$32 \times 10^{-6} / 0{,}2 * 100 = 0{,}016\ \%$

Per augmentar la velocitat de transferència i disminuir el temps que la CPU dedica a entrades des de teclat podríem fer transferència de bloc suposant que el teclat té un buffer per guardar un grup de tecles, que serien transmeses en una sola atenció d'interrupció. D'aquesta manera, el temps invertit

en totes les feines auxiliars del tractament de les interrupcions (que és el temps que enlenteix el procés) només afectaran un cop a cada transferència de bloc. Però, per altra banda, aquesta estratègia faria que el sistema no respongués a l'entrada de cada tecla introduïda sinó quan hi hagués un grup de tecles disponibles. Per tant, si pensem en la utilitat normal d'un teclat, veiem que aquest funcionament no és adequat. Altres solucions que no tinguessin aquests desavantatges passen per tècniques de DMA o canals d'entrada/sortida.

Problema 28

A)

$f = 5$ MHz => $T = 0{,}2$ µs

Temps per transferir el control dels busos, $T_B = 15 \times 0{,}2 = 3$ µs

Temps per fer una transferència senzilla per DMA, $T_{DMA} = 25 \times 0{,}2 = 5$ µs

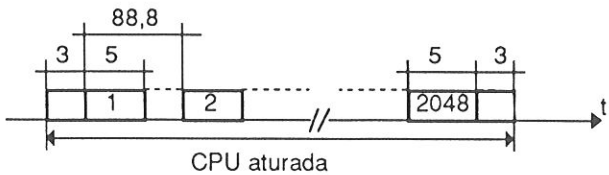
La velocitat de transferència del disc serà:

(Densitat de bits/pista) / (velocitat de rotació) =

= 15000 (bits/pista) x (360/60) (voltes/s) = 90000 bits/s = 11250 Bytes/s

Per tant: període de transferència disc, $T_d = 1 / 11250 = 88{,}8$ µs (un byte cada 88,8 µs).

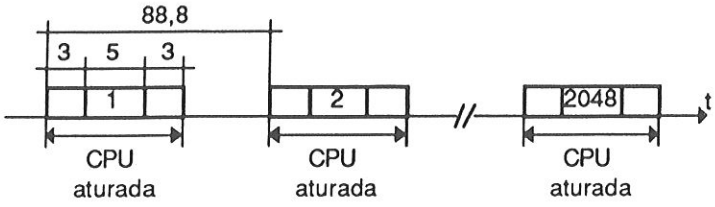
Si féssim la transferència en mode burst:



Temps desocupació CPU = $2047 \times T_d + T_{DMA} + 2 \times T_B =$

$= 2047 \times 88{,}8 + 5 + 2 \times 3 = 0{,}1817846$ s

Si féssim la transferència en mode byte:



Temps desocupació CPU = 2048 x (T_{DMA} + 2 x T_B) = 2048 x (5 + 2 x 3) =

= 0,022528 s

Per tant, donat que la velocitat de transferència serà la mateixa pels dos modes, el mode byte és l'adequat ja que dóna un temps menor de desocupació de CPU.

B)

La capacitat del disc serà:

C = 15000 (bits/pista) x 2 (cares) x 80 (pistes) = 2400000 bits =

= 300000 B = 293 KB

Tests

34013100				34013200		34013300		34013400	
1.	A	16.	B	1.	D	1 -	B	1 -	B
2.	B	17.	D	2.	C	2 -	D	2 -	D
3.	D	18.	B	3.	D	3 -	C	3 -	A
4.	C	19.	C	4.	B	4 -	A	4 -	C
5.	B	20.	C	5.	D	5 -	B	5 -	D
6.	D	21.	B	6.	D	6 -	D	6 -	B
7.	D	22.	D	7.	D	7 -	D	7 -	D
8.	B	23.	D	8.	A	8 -	A	8 -	A
9.	B	24.	C	9.	D	9 -	C	9 -	D
10.	A	25.	C	10.	A	10 -	A	10 -	B
11.	D	26.	A	11.	C	11 -	C	11 -	D
12.	C	27.	D	12.	B	12 -	A	12 -	C
13.	A	28.	D	13.	A	13 -	D	13 -	C
14.	D	29.	C	14.	A	14 -	A	14 -	A
15.	A	30.	D	15.	C	15 -	C	15 -	D
				16.	A	16 -	D	16 -	B
				17.	B	17 -	C	17 -	A
				18.	D	18 -	D	18 -	A
				19.	A	19 -	A	19 -	B
				20.	D	20 -	D	20 -	C

340130500

1 -	D
2 -	A
3 -	C
4 -	D
5 -	A
6 -	B
7 -	D
8 -	A
9 -	D
10 -	D
11 -	B
12 -	B
13 -	C
14 -	A
15 -	B
16 -	D
17 -	B
18 -	A
19 -	D
20 -	C

3407401306000

1 -	A
2 -	B
3 -	C
4 -	C
5 -	B
6 -	B
7 -	ANUL.LADA
8 -	D
9 -	A
10 -	B
11 -	C
12 -	A
13 -	A
14 -	C
15 -	A
16 -	C
17 -	C
18 -	B
19 -	A
20 -	B

— CPU: execució de programes

- ALU
- Registres
- Unitat de Control
  - CABLEJADA: descodificació a nivell de hard.
  - MICROPROGRAMADA: instruccions en codi màquina. Instruccions binàries. microinstruccions. Modificat d'instruccions.

— MAQUINES

Pipeline — estructura sèrie

Paral·lel: — SISD — 1 UCP i 1 memòria
— SIMD — 1 memòria, molts mitjans de tractament, 1 unitat de Control.
— MIMD — " " , molts UT i molts UC.

Simples / multiples Instruccions Simples / Multiples Dades

RISC	CISC
Instruccions senzilles	Instruccions Complexes
facilitat a PIPELINE	Poca PIPELINE
Pocques Instruccions i Modes	Moltes Instruccions i Modes
Complexitat en el compilador	Complexitat en el microprograma

— 8086

AX — acumulador 8bits + 8bits
BX — base
CX — contador
DX — dades

Base — SP, BP
DI — Index destí
SI — Index font

Segments — CS, DS, ES, SS.

U.C. microprogramada.

Modes — Mínim: Senyal de control.
Màxim: Més senyals de control. Coprocessador.

Adreça física: $\boxed{\begin{matrix} DS \\ CS \end{matrix} \times 16 + IP}$

— memòria — principal: RAM, ROM
— secundària: discos, ...

Idea
Com estàs fa?